UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

A TOOL FOR ONTOLOGY INSTANCE MATCHING

André Filipe Agostinho Lopes

DISSERTAÇÃO

MESTRADO EM INFORMÁTICA

2013

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

A TOOL FOR ONTOLOGY INSTANCE MATCHING

André Filipe Agostinho Lopes

DISSERTAÇÃO

MESTRADO EM INFORMÁTICA

Dissertação orientada pelo Prof. Doutor Francisco José Moreira Couto
e co-orientado pelo Prof. Doutor Mário Jorge Costa Gaspar da Silva

2013

Acknowledgments

I would like to thank my advisors, Dr. Francisco Couto and Dr. Mário Silva, for their guidance. I also thank the REACTION group, where it is included Dr. Mário Silva, because through their criticisms, in our periodic meetings, helped me to improve my work, and within the group, to Sílvio Moreira for his technical guidance that was crucial in the beginning of my dissertation.

Finally, I am very grateful to LaSIGE for providing me a good place to work, and also to the Portuguese *Fundação para a Ciência e a Tecnologia* through the financial support of the SOMER project (PTDC/EIA-EIA/119119/2010) that gave me a scholarship for conducting my studies. At last, to my parents and grandmother, that gave me emotional support.

Thank You Life for the ride

Resumo

A Web Semântica pretende fornecer formatos comuns para caracterizar semanticamente os dados publicados na Web, melhorando a interoperabilidade e integração de dados. A iniciativa *Linked Data* visa ligar dados relacionados que não foram previamente ligados. As ontologias têm um papel fundamental nisso, pois, fornecem vocabulários controlados, para caracterizar semanticamente os dados de uma forma inequívoca.

Conforme definido por Gruber, uma ontologia é uma especificação de uma conceituação, que se destina a modelar um domínio em particular. A especificação de uma ontologia é composto por dois tipos de declarações: TBox (classes) e ABox (exemplares). **TBox** são classes que são interpretadas como um conjunto de indivíduos no domínio; **ABox** são exemplares que são interpretados como indivíduos particulares de um domínio. Além disso, uma ontologia também é composta por: **Relacionamentos** ou relações entre classes e/ou exemplares; **Tipos de dados** são partes particulares do domínio que especificam valores; **Valores de dados** são valores simples.

Apesar de uma ontologia se destinar a modelar um domínio em particular, existem muitas ontologias de diferentes fontes a modelar o mesmo domínio, isto é, existe um problema de sobreposição. O problema de sobreposição consiste em ontologias distintas que representam as mesmas entidades de uma forma diferente. É, portanto, necessário criar processos capazes de encontrar as sobreposições e fundi-las.

Emparelhamento de ontologias é geralmente aplicado para alinhar duas TBox de duas ontologias diferentes, ou seja, para encontrar relações ou correspondências entre as classes ontológicas.

Há um caso particular de emparelhamento de ontologias, o Emparelhamento de Exemplares. O objetivo do emparelhamento de exemplares é alinhar dois ABox de duas ontologias diferentes, ou seja, encontrar as correspondências entre exemplares de diferentes ontologias. O Emparelhamento de Exemplares adota o princípio de que, quanto maior for a semelhança entre duas descrições de exemplares de duas ontologias distintas, maior é a probabilidade de estes exemplares representarem a mesma entidade de um determinado domínio. Por exemplo, no domínio político, vamos considerar o actual Presidente da Comissão Europeia, Durão Barroso e assumir que na Ontologia 1 tem um exemplar com o descritor: “José Manuel Durão Barroso”, e Ontologia 2 tem um exemplar com o descritor: “José Durão Barroso”. Portanto, é necessário implementar técnicas de emparelhamento

de exemplares, para descobrir se estes dois exemplares destas duas ontologias diferentes correspondem à mesma pessoa/entidade, isto é, se eles emparelham.

Os objectivos desta dissertação eram:

Desenvolvimento de algoritmos de emparelhamento de exemplares Que visou o desenvolvimento de algoritmos para o emparelhamento de ontologias ao nível dos seus exemplares, de forma a resolver problemas de emparelhamento de exemplares. O desenvolvimento de algoritmos foi baseado em técnicas de emparelhamento de exemplares já propostas por outros;

Alinhamento de exemplares do mundo real Que visou a aplicação dos algoritmos desenvolvidos, para gerar emparelhamentos de alta qualidade em exemplares do mundo real, e avaliar a sua qualidade em termos de Precisão, Sensibilidade, Medida-F, Exatidão e Exatidão Unilateral;

Desenvolvimento de um emparelhador de exemplares Web Que visou o desenvolvimento de uma ferramenta capaz de realizar emparelhamento de exemplares através da Web, incorporando os algoritmos desenvolvidos por mim.

Os resultados alcançados por esta dissertação foram a produção de alinhamentos de exemplares, entre as ontologias POWER-DBpediaPT, POWER-Verbetes e POWER-POWER. Estas três ontologias contêm exemplares que representam entidades políticas. E também entre as ontologias provenientes do OAEI 2012. O OAEI (***O**ntology **A**lignment **E**valuation **I**nitiative*), é um concurso international, realizado todos os anos, que entre vários tipos de competições, tem uma dedicada à avaliação de ferramentas e de técnicas de emparelhamento de exemplares. Para avaliar a qualidade dos alinhamentos produzidos foram implementadas as seguintes métricas: Precisão; Sensibilidade; Medida-F; Exatidão; e Exatidão Unilateral. Esta dissertação também produziu um emparelhador de exemplares disponível através da Web, que implementa as métricas mencionadas para avaliar os alinhamentos produzidos por ele.

POWER (***P**olitics **O**ntology for **W**eb **E**ntity **R**etrieval*) é uma ontologia que modela o domínio da política Portuguesa, que foi desenvolvida e fornecida pela grupo REACTION. Os seus exemplares foram alinhados com os das ontologias DBpediaPT e Verbetes. A DBpediaPT é uma ontologia que contém exemplares que representam entidades da DBpedia versão 3.8. Cada entidade é referida na versão Portuguesa da Wikipedia. Esta ontologia foi construída a partir de uma lista, fornecida pelo grupo REACTION. Verbetes é uma ontologia, cujos os exemplares representam entidades que têm pelo menos cinco ocorrências nas notícias agregadas pelo serviço SAPO Verbetes.

Para avaliar o alinhamento POWER-DBpediaPT foi usada a métrica Exatidão Unilateral. Usando o algoritmo de emparelhamento *FirstLastNamePlusJaccard*, alcançou-se 97.29% de Exatidão Unilateral para o POWER, e 87.25% de Exatidão Unilateral para o

DBpediaPT. Usando o algoritmo de emparelhamento *Stratified 10-fold Cross-Validation*, alcançou-se 99.11% de Exatidão Unilateral para o POWER, e 95.97% de Exatidão Unilateral para o DBpediaPT. Estes foram os melhores resultados consigos para este alinhamento. No caso do alinhamento POWER-Verbetes não foram calculadas métricas mas, fez-se uma avaliação manual pela minha parte e pela parte do grupo REACTION, e foi positiva. Além disso, porque o POWER contém exemplares duplicados, ou seja, dois ou mais exemplares a representarem a mesma entidade, foi efectuado o alinhamento POWER-POWER de forma a encontrar os exemplares duplicados. No caso do POWER, estas situações não podiam acontecer. O alinhamento foi entregue ao grupo REACTION, para eles poderem melhorar a sua ontologia. Estes dois alinhamentos, POWER-Verbetes e POWER-POWER, foram realizados pelo algoritmo de emparelhamento *MachineLearning*.

Foram também realizados alinhamentos de exemplares entre as ontologias fornecidas pelo OAEI 2012. Estas ontologias encontram-se divididas em dois grupos: o *Sandbox* que contém onze ontologias; e o *IIMB* que contém oitenta ontologias. Os alinhamentos produzidos foram realizados dentro de cada grupo. Neste caso, os algoritmos de emparelhamento utilizados foram *FirstLastNamePlusJaccard* e o *Stratified 10-fold Cross-Validation*. Na maioria dos alinhamentos produzidos a Medida-F foi maior no segundo algoritmo do que no primeiro.

Todas as ontologias cujos os exemplares foram alinhados, e os seus respectivos alinhamentos e métricas, estão disponíveis através da ligação: `http://lasige.di.fc.ul.pt/webtools/instancematcher/dissertation_work.zip`.

O emparelhador de exemplares Web, foi outra realização desta dissertação, e está disponível através da ligação: `http://lasige.di.fc.ul.pt/webtools/instancematcher/`. Este disponibiliza aos utilizadores dois algoritmos de emparelhamento: o *FirstLastNamePlusJaccard*; e o *MachineLearning*. Além disso, também permite que o utilizador escolha que tipo de alinhamentos quer. Um-para-um (em Inglês: *one-to-one*) ou muitos-para-muitos (em Inglês: *many-to-many*). No primeiro caso, cada exemplar só pode estar presente uma vez no alinhamento, isto é, não pode haver mais do que um emparelhamento por exemplar; no segundo caso, cada exemplar pode estar presente várias vezes no alinhamento, ou seja, pode haver mais do que um emparelhamento por exemplar. Os alinhamentos POWER-DBpediaPT e POWER-Verbetes foram um-para-um. E os alinhamentos OAEI 2012 e POWER-POWER foram muitos-para-muitos. Há ainda a opção *Limiar* (em Inglês: *Threshold*) que permite ao utilizador indicar qual é o valor mínimo dos alinhamentos devolvidos pelo emparelhador de exemplares Web. Em cada alinhamento de exemplares é atribuído um valor [0,1] pelos algoritmos de emparelhamento, que determina o grau de confiabilidade/certeza do alinhamento estabelecido. No alinhamento também se podem encontrar exemplares que emparelham para nada, ou seja, para NULL. Estes, são os exemplares para os quais o algoritmo de emparelhamento escolhido, não

encontrou nenhum exemplar correspondente. Para que o emparelhador de exemplares Web devolva métricas que atestem a qualidade do alinhamento produzido, o utilizador tem que introduzir o alinhamento de referência (em Inglês: *Reference Alignment*). Este é um documento, que se assume, que contenha todos os emparelhamentos correctos entre os exemplares de duas ontologias. As métricas são calculadas aquando da comparação do alinhamento produzido com o alinhamento de referência. Existem ainda as opções POWER 2010 e OAEI 2012, que permitem indicar ao emparelhador de exemplares Web, que os exemplares a emparelhar são do POWER e do OAEI 2012. É também necessário que o utilizador insira os identificadores dos descritores dos exemplares, para que o emparelhador obtenha a informação necessária para poder efectuar os alinhamentos. Cada identificador tem que começar pelo prefixo *http*.

Palavras-chave: Web Semântica, Ontologias, Emparelhamento de Ontologias, Emparelhamento de Exemplares, Emparelhamento de cadeia de caracteres, Aprendizagem Automática

Abstract

An ontology is an object-based conceptualization of some particular domain. An ontology provides a shared controlled vocabulary to semantically characterize the data of the modelled domain. But it often happens that independently created ontologies model the same domain in different ways. This constitutes a problem because there may be entities being represented differently, therefore creating ambiguity and interoperability problems when linking related data characterized by two ontologies. So it is necessary to develop processes capable of matching the data.

The matching can be made at the class level or at the instance level. The goal of the instance matching is to find the correspondences between instances from different ontologies, called instance alignments.

The objective of this dissertation was the development of instance matching algorithms for generating instance alignments of real world instances. And the creation of an instance matcher Web tool, where the algorithms developed by me were incorporated.

The outcome of this dissertation was the generation of instance alignments between POWER-DBpediaPT, POWER-Verbetes and POWER-POWER. All these three ontologies have instances representing political entities. Furthermore, it was generated instance alignments between ontologies from the OAEI 2012. OAEI (**O**ntology **A**lignment **E**valuation **I**nitiative), is an international contest, that has a track focus on evaluation of instance matching tools and techniques. To assess the quality of the instance alignments produced, it was implemented the metrics of Precision, Recall, F-measure, Accuracy and Unilateral Accuracy.

Another outcome of this dissertation is the instance matcher tool, available through the Web. The tool implements two instance matchers. The FirstLastNamePlusJaccard which is based on element-level matching techniques, that uses the descriptors of the instances to correspond them. And the MachineLearning matcher that uses machine learning approaches to find those correspondences. This Web tool also assesses the instance alignments that it produces, because it implements the already mentioned metrics.

Keywords: Semantic Web, Ontologies, Ontology Matching, Instance Matching, String matching, Machine Learning

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The Semantic Web intends to provide common formats for semantically characterizing data published on the web, improving interoperability and integration of data (Berners-Lee et al., 2001). The *Linked Data* initiative aims at connecting related data that wasn't previously linked (Bizer et al., 2009). Ontologies have a crucial role on this, since they provide shared controlled vocabularies to semantically characterize the data in an unambiguous way.

As defined by Gruber, an ontology is a specification of a conceptualization (Gruber, 2008), that is meant to model some particular domain. The specification of an ontology is composed by two types of statements: TBox (classes) and ABox (instances). **TBox** are classes which are interpreted as a set of individuals in the domain; **ABox** are instances which are interpreted as particular individuals of a domain. Moreover, an ontology is also composed by: **Relationships** or relations between classes and/or instances; **Data types** are particular parts of the domain which specify values; **Data values** are simple values (Euzenat and Shvaiko, 2007).

Although an ontology is meant to model some particular domain, there are many ontologies from different sources modelling the same domain, i.e., there is an overlapping problem. The overlapping problem consists on distinct ontologies representing the same entities in a different manner. It is thus necessary to create processes capable of finding those overlaps and merge them.

Ontology matching is usually applied to align two TBox from two different ontologies, i.e., to find relationships or correspondences between ontological classes. These correspondences are the relation holding, or supposed to hold according to a particular matching algorithm, between classes of different ontologies (Euzenat and Shvaiko, 2007).

There is a particular case of ontology matching, the **Instance Matching**. The goal of instance matching is to align the two ABox from two different ontologies, i.e., to find the correspondences between instances of different ontologies. An instance matching will

adopt the principle that the higher is the similarity between two instance descriptions of two distinct ontologies, the higher is the probability of these instances represent the same entity in a given domain. For example, in the political domain, let us consider the current President of the European Commission, Durão Barroso and assume that Ontology 1 has an instance with the label: "José Manuel Durão Barroso"; and Ontology 2 an instance with a label: "José Durão Barroso". Therefore, it is necessary to implement instance matching techniques to find if these two instances of these two different ontologies correspond to the same entity, i.e., if they match. Another example, is the POWER, DBpediaPT and Verbetes ontologies that have instances representing political entities.

1.2 Goals

The objectives of this work were:

Development of instance matching algorithms That aimed at developing algorithms for ontology matching at the instance level, in order to solve instance matching problems. The development of algorithms was based on current instance matching techniques already proposed by others;

Real world instance alignments That aimed at applying the algorithms developed to generate high quality matches for real world instances and assess their quality in terms of Precision, Recall, F-measure, Accuracy and Unilateral Accuracy;

Instance matcher Web tool development That aimed at developing a tool capable of performing instance matching through the Web, incorporating the algorithms developed by me.

1.3 Contributions

The contributions of this work are the following:

- Development of instance matching algorithms;
- Real world instance alignments, namely: POWER-DBpediaPT and POWER-Verbetes;
- POWER-POWER alignment to help the POWER developers to improve it;
- The creation of a Instance Matcher tool, available through the Web.

1.4 Planning

Date	Milestones
1^{st} of October 2012	Familiarization with the work nature
5^{th} of November 2012	First set of instance matching algorithms, Reference alignment and Preliminary report
1^{st} of December 2012	First Tool Prototype, and Reference alignment update
1^{st} of January 2013	Real world instance alignments
1^{st} of February 2013	Second set of instance matching algorithms
1^{st} of March 2013	Second Tool Prototype, and Reference Alignment update
1^{st} of April 2013	Real world instance alignments
1^{st} of May 2013	Master thesis and OAEI 2013 participation
1^{st} of June 2013	Article about the obtained results

Table 1.1: The milestones set in the Preliminary report

The Table 1.1 presents the milestones of the original planning, set for this dissertation in the preliminary report. There was some deviations from the original milestones. The original planning supposed an iteration workflow that did not occur, because during the term of this work, in practice, it did not justify. The OAEI 2013 participation did not happen and an article about the obtained results was not made, due to the additional work caused by the POWER-DBpediaPT alignment. This also contributed to the delay in the delivery of the dissertation report. This milestone suffered a delay of almost 5 months.

Although the OAEI 2013 participation did not occur, it was performed instance alignments between the ontologies provided by the OAEI 2012, for preparation. OAEI (**O**ntology **A**lignment **E**valuation **I**nitiative) (Ehrig and Sure, 2005) is an international contest, held every year, that has a track focus on evaluation of instance matching tools and techniques.

1.5 Document Structure

This document is structured as follows:

Chapter 2 – Presents some concepts and subjects necessary to understand this work;

Chapter 3 – Describes the architecture of the system developed and its modules;

Chapter 4 – Presents the results obtained by the system;

Chapter 5 – Shows conclusions and future work.

This work was done in the scope of the *Master in Informatics* provided by the *Department of Informatics* of the *Faculty of Sciences of University of Lisbon*. This work also occurred within the SOMER project (PTDC/EIA-EIA/119119/2010), that is financially supported by the Portuguese *Fundação para a Ciência e a Tecnologia*.

Chapter 2

Related Work

This chapter presents some concepts and subjects necessary to understand this work and, it also introduces available instance matching resources and techniques that were used in this dissertation.

2.1 Ontologies

In the following sections it is presented the ontologies from where it was performed the instance alignments. It is also presented the sources from where the ontologies are from.

2.1.1 OAEI 2012

The **O**ntology **A**lignment **E**valuation **I**nitiative (Ehrig and Sure, 2005), held every year, it is an international contest, that among different kinds of competitions, has a track focus on evaluation of instance matching tools and techniques. That evaluation relates to the outcome of these tools and techniques, i.e., the alignments produced by them. The quality of these alignments are evaluated by comparing them with predefined reference alignments, provided by the contest, which are used to produce Precision, Recall and F-measure metrics.

The dissertation focuses on two tasks of the OAEI 2012[1] instance matching track: the Sandbox task which is composed by eleven ontologies, and contains light matching problems such as, labels containing light textual changes; and the IIMB task which is composed by eighty ontologies, and contains hard matching problems such as, strong textual changes, and stronger structural and logical transformations. This track has a set of rules to follow: the alignments produced must be between two ontologies, and one of them must be the reference ontology. For example, the Sandbox or IIMB reference ontologies must be matched against each of the ontologies that respectively compose the Sandbox and IIMB tasks. The alignments produced are many-to-many, i.e., one instance

[1] http://oaei.ontologymatching.org/2012/

from one ontology can be matched to multiple instances of the other ontology. Finally, the alignments must respect the .RDF OAEI 2012 format.

<owl:NamedIndividual rdf:about="http://oaei.ontologymatching.org/2012/IIMBDATA/en/andy_secombe">

<rdf:type rdf:resource="http://oaei.ontologymatching.org/2012/IIMBTBOX/Actor"/>

<IIMBTBOX:date_of_birth rdf:datatype="http://www.w3.org/2001/XMLSchema#string">1953-04-26</IIMBTBOX:date_of_birth>

<IIMBTBOX:article rdf:datatype="http://www.w3.org/2001/XMLSchema#string"><p>Andrew Secombe (born 26 April 1953 in Mumbles, south Wales), better known as Andy Secombe, is a Welsh actor, voice actor, and author. He played Rover the Dog in the Channel 4 children's series Chips Comic. Son of comedian/singer Harry Secombe (whom he later impersonated in a Goon Show special), the younger Secombe is probably best known for providing the voice of Watto in the Star Wars prequels. Andy provided the voice of another Toydarian in Star Wars: Knights of the Old Republic II: The Sith Lords. He subsequently shifted to writing, penning four fantasy novels to date: Limbo; Limbo Two: The Final Chapter; The Last House in the Galaxy; and Endgame. In 2005, he returned to acting, becoming part of another sci-fi franchise by playing Colin the Security Robot in The Hitchhiker's Guide to the Galaxy: Quintessential Phase.</p>

</IIMBTBOX:article>

<IIMBTBOX:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Andy Secombe</IIMBTBOX:name>

<IIMBTBOX:gender rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Male</IIMBTBOX:gender>

</owl:NamedIndividual>

Figure 2.1: Sample instance from the Sandbox and IIMB reference ontology

<owl:NamedIndividual rdf:about="http://oaei.ontologymatching.org/2012/IIMBDATA/en/item6178572603824285307">

<rdf:type rdf:resource="http://oaei.ontologymatching.org/2012/IIMBTBOX/Actor"/>

<IIMBTBOX:date_of_birth rdf:datatype="http://www.w3.org/2001/XMLSchema#string">1953-04-26</IIMBTBOX:date_of_birth>

<IIMBTBOX:article rdf:datatype="http://www.w3.org/2001/XMLSchema#string"><p>Andrew Secombe (born 26 April 1953 in Mumbles, south Wales), better known as Andy Secombe, is a Welsh actor, voice actor, and author. He played Rover the Dog in the Channel 4 children's series Chips Comic. Son of comedian/singer Harry Secombe (whom he later impersonated in a Goon Show special), the younger Secombe is probably best known for providing the voice of Watto in the Star Wars prequels. Andy provided the voice of another Toydarian in Star Wars: Knights of the Old Republic II: The Sith Lords. He subsequently shifted to writing, penning four fantasy novels to date: Limbo; Limbo Two: The Final Chapter; The Last House in the Galaxy; and Endgame. In 2005, he returned to acting, becoming part of another sci-fi franchise by playing Colin the Security Robot in The Hitchhiker's Guide to the Galaxy: Quintessential Phase.</p>

</IIMBTBOX:article>

<IIMBTBOX:gender rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Male</IIMBTBOX:gender>

<IIMBTBOX:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Secombe, A.</IIMBTBOX:name>

</owl:NamedIndividual>

Figure 2.2: Sample instance from the Sandbox 001 ontology

```
<owl:NamedIndividual rdf:about="http://oaei.ontologymatching.org/2012/IIMBDATA/en/item644971858866295946">
    <rdf:type rdf:resource="http://oaei.ontologymatching.org/2012/IIMBTBOX/Actor"/>
    <IIMBTBOX:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">A. S.</IIMBTBOX:name>
    <IIMBTBOX:date_of_birth rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Apr 26, 1953</IIMBTBOX:date_of_birth>
    <IIMBTBOX:gender rdf:datatype="http://www.w3.org/2001/XMLSchema#string">M</IIMBTBOX:gender>
    <IIMBTBOX:article rdf:datatype="http://www.w3.org/2001/XMLSchema#string">some Secombe (born some Apr Phase.&lt;/p&gt; Lords. Mumbles, to of 2005, prequels. better 26 novels Andy up leading a war played Colin actor, and author. in actor, to the chase in voice comedian/singer he credibly some south The 26 of put harry the Two: he he automaton in type acting, hood to to Toydarian 2005, Secombe Star Endgame. (born Secombe war provide Secombe, the sound Watto The of show Watto Wales), Andy provide return cost the &lt;p&gt;Andrew other acting, in of Wars: knight the the to republic II: The Sith equally Secombe later switch to cost up unseasoned fantasy becoming special), date: republic limbo of war last Chapter; Comic. cost sci-fi the novels credibly Galaxy; and 26 4 knap Andy the sci-fi automaton in partially known series to transmit franchise 4 play sound the sci-fi pose voice Son Hitchhiker&#39;s guide of Secombe, in quintessential 1953
  </IIMBTBOX:article>
  </owl:NamedIndividual>
```

Figure 2.3: Sample instance from the IIMB 018 ontology

```
<map>
  <Cell>
    <entity1 rdf:resource="http://oaei.ontologymatching.org/2012/IIMBDATA/en/andy_secombe"/>
    <entity2 rdf:resource="http://oaei.ontologymatching.org/2012/IIMBDATA/en/item644971858866295946"/>
    <relation>=</relation>
    <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1.0</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource="http://oaei.ontologymatching.org/2012/IIMBDATA/en/brian_blessed"/>
    <entity2 rdf:resource="http://oaei.ontologymatching.org/2012/IIMBDATA/en/item2598975824331664252"/>
    <relation>=</relation>
    <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1.0</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource="http://oaei.ontologymatching.org/2012/IIMBDATA/en/windhoek"/>
    <entity2 rdf:resource="http://oaei.ontologymatching.org/2012/IIMBDATA/en/item6541897249775690306"/>
    <relation>=</relation>
    <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1.0</measure>
  </Cell>
</map>
```

Figure 2.4: Predefined reference alignment sample

The Figure 2.1 shows that the reference ontologies of both Sandbox and IIMB are duplicates, i.e., have the same type of instances. The difference relies on the ontologies that compose the Sandbox and IIMB tasks. As shown in the Figure 2.3, the properties *name*, *date_of_birth* and *gender* have suffered more changes in their content than in the instance shown in the Figure 2.2. These three figures represent three different ways of describing the same entity. The entity is this case, is an actor called *Andrew Secombe*

born in 26th of April 1953, belonging to the male gender, and with the profession of actor. More details about the entity is then given by the *article* property.

The Figure 2.4 shows a sample of a predefined reference alignment, provided by the contest, between the IIMB reference ontology and the IIMB 018 ontology. Each *map* tag contains within: the pair of instances that are possible to be match, represented by their unique identifiers; the relation between them; and the confidence score of the match, represented by the *measure* tag. The first *map* tag of the example, shows that the instance present in the Figure 2.1 and the instance present in the Figure 2.3 are to be matched, because they are referring to the same entity. The alignments produced must be in the format shown in the Figure 2.4.

The OAEI **2012** instance matching track is no longer available, because it has been replaced by the OAEI **2013** instance matching track version. The *Sandbox* and *IIMB* group of ontologies of the OAEI 2012 can be accessed through this link: `http://lasige.di.fc.ul.pt/webtools/instancematcher/dissertation_work.zip`

2.1.2 POWER

POWER - Politics Ontology for Web Entity Retrieval - is an ontology that models the Portuguese political domain, that was built and provided by the REACTION group[2]. More precisely, POWER is an ontology of political processes and entities. It is designed for tracking politicians, political organizations and elections, both in mainstream and social media (Moreira et al., 2012). POWER is an ontology which needs yet to be improved, because it has duplicate instances. This situation in POWER, is considered not to be correct. It is available in the following site:`http://dmir.inesc-id.pt/project/POWER-PT_01_in_English`.

```
<power:Politician rdf:about="http://dmir.inesc-id.pt/pub/publications/2010/power#paulo_sacadura_cabral_portas_merged">
    <power:mergeFrom rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#paulo_portas"/>
    <power:mergeFrom rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#paulo_sacadura_cabral_portas"/>
    <power:hasAffiliation rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#paulo_sacadura_cabral_portas_affto_c-pp"/>
    <power:isReferredBy rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#paulo_portas_birthname"/>
    <power:isReferredBy rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#paulo_sacadura_cabral_portas_birthname"/>
    <power:servesMandate rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#deputado_da_assembleia_da_republica_1999"/>
    <power:servesMandate rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#ministro_da_defesa_nacional_2002"/>
    <power:servesMandate rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#deputado_da_assembleia_da_republica_1995"/>
    <power:servesMandate rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#ministro_da_defesa_nacional_e_dos_assuntos_do_mar_2004"/>
    <power:servesMandate rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#deputado_da_assembleia_da_republica_2009"/>
    <power:servesMandate rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#deputado_da_assembleia_da_republica_2005"/>
    <power:servesMandate rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#ministro_de_estado_2002"/>
    <power:servesMandate rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#ministro_dos_negocios_estrangeiros_2012"/>
    <power:servesMandate rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#ministro_de_estado_2012"/>
    <power:servesMandate rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#ministro_de_estado_2004"/>
    <power:servesMandate rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#deputado_da_assembleia_da_republica_2002"/>
</power:Politician>
```

Figure 2.5: Sample instance from POWER. In bold, the labels' references. The labels were the only data from POWER that were used on this dissertation.

[2]http://dmir.inesc-id.pt/project/Reaction

```
<power:EntityName rdf:about="http://dmir.inesc-id.pt/pub/publications/2010/power#paulo_portas_birthname">
  <power:name>PAULO PORTAS</power:name>
  <power:hasType rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#BirthName"/>
  <power:refers rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#paulo_portas"/>
  <power:refers rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#paulo_sacadura_cabral_portas_merged"/>
 </power:EntityName>

<power:EntityName rdf:about="http://dmir.inesc-id.pt/pub/publications/2010/power#paulo_sacadura_cabral_portas_birthname">
  <power:name>PAULO SACADURA CABRAL PORTAS</power:name>
  <power:hasType rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#BirthName"/>
  <power:refers rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#paulo_sacadura_cabral_portas_merged"/>
  <power:refers rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#paulo_sacadura_cabral_portas"/>
</power:EntityName>
```

Figure 2.6: Sample labels from POWER. In bold, the labels' content from the instance of the Figure 2.5.

2.1.3 DBpediaPT

DBpediaPT is an ontology that contains instances that represent people from the DBpedia version 3.8. Each person is referred in the Portuguese version of Wikipedia. This ontology was built from the Person list[3], provided by the REACTION group. I took the list and converted it into an ontological format (.NT).

```
<http://pt.wikipedia.org/wiki/Albino_Forjaz_de_Sampaio> <> <> .
<http://pt.wikipedia.org/wiki/Albert_Einstein> <> <> .
<http://pt.wikipedia.org/wiki/Adriano> <> <> .
<http://pt.wikipedia.org/wiki/Afonso,_Príncipe_de_Portugal_(1475-1491)> <> <> .
<http://pt.wikipedia.org/wiki/Alexandre_Rodrigues_Ferreira> <> <> .
<http://pt.wikipedia.org/wiki/Aldous_Huxley> <> <> .
<http://pt.wikipedia.org/wiki/Aleksandr_Oparin> <> <> .
<http://pt.wikipedia.org/wiki/Antônio_Mariano_de_Oliveira> <> <> .
<http://pt.wikipedia.org/wiki/António_Guterres> <> <> .
```

Figure 2.7: Sample instances from DBpediaPT

Supporting ontology

The supporting ontology is of the same version of the DBpedia (3.8), and it is available in the following site:`http://downloads.dbpedia.org/3.8/pt/`. This ontology contains abstract texts for each person of the Portuguese version of Wikipedia, giving several types of information such as the complete name of the person, its nationality, and profession.

[3] http://dmir.inesc-id.pt/project/DBpediaEntities-PT_01_in_English

<http://pt.dbpedia.org/resource/Durão_Barroso> <http://dbpedia.org/ontology/abstract> "**José Manuel Durão Barroso é um político e professor português**, actual presidente da Comissão Europeia, cargo que ocupa desde Novembro de 2004. Em Portugal, foi sub-secretário do ministério dos assuntos internos, em 1985, e ministro dos Negócios Estrangeiros em 1992. Entre 2002 e 2004, ocupou o cargo de primeiro-ministro da República Portuguesa. A 23 de novembro de 2004, Durão Barroso assumiu as funções de Presidente da Comissão Europeia, cargo que irá assumir outra vez em Novembro de 2009, após ter sido reeleito pelo Parlamento Europeu a 16 de Setembro."@pt .

Figure 2.8: Sample instance from the Supporting ontology. In bold, the complete name, the nationality and the profession, of the person represented in the instance.

2.1.4 Verbetes

Verbetes is an ontology, which instances represent people that have at least five occurrences in the news aggregated by the SAPO Verbetes service[4]. I took the list of people returned by the service and converted it into an ontological format (.NT).

<http://services.sapo.pt/InformationRetrieval/Verbetes/WhoIs?name=Jorge Rosa> <> <> .
<http://services.sapo.pt/InformationRetrieval/Verbetes/WhoIs?name=David Dinis> <> <> .
<http://services.sapo.pt/InformationRetrieval/Verbetes/WhoIs?name=Michael Smith> <> <> .
<http://services.sapo.pt/InformationRetrieval/Verbetes/WhoIs?name=Edy Reja> <> <> .
<http://services.sapo.pt/InformationRetrieval/Verbetes/WhoIs?name=Robert Mueller> <> <> .
<http://services.sapo.pt/InformationRetrieval/Verbetes/WhoIs?name=Sigmar Gabriel> <> <> .
<http://services.sapo.pt/InformationRetrieval/Verbetes/WhoIs?name=Daniel Gonçalves> <> <> .
<http://services.sapo.pt/InformationRetrieval/Verbetes/WhoIs?name=Coelho Gil> <> <> .

Figure 2.9: Sample instances from Verbetes

2.2 Element-level matching techniques

Terminological approaches (Couto and Pinto, 2013) focus on the descriptors of the classes or of the instances, as opposed to structural approaches that explore the structure of the classes or of the instances, i.e., their relations to other classes or to other instances. The element-level matching techniques encompass several techniques, although only string-based techniques, were used in this dissertation.

String-based techniques (Euzenat and Shvaiko, 2007) are often used in order to match names of ontology entities. They are typically based on the following intuition: the more similar the names/strings, the more likely they denote the same entities. This dissertation used the Jaccard Similarity Coefficient (Jaccard, 1912) string-based technique, which given two strings (for example, instance labels) produces a (confidence) score [0,1] that represents the degree of similarity between them. Higher is the score, higher is the probability of these instances correspond to the same entity. This technique measures similarity

[4]http://softwarelivre.sapo.pt/projects/developers/wiki/Services/InformationRetrieval/Verbetes_EN

between strings through the size of the intersection divided by the size of the union of the words composing the strings:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

Figure 2.10: Jaccard formula
Source: http://en.wikipedia.org/wiki/Jaccard_index

For example, A = "José Durão Barroso"; B = "Durão Barroso" = |"Durão"∩"Barroso"|/ |"José"∪ "Durão"∪"Barroso"|= 2/3 = 0.66.

2.3 Machine Learning techniques

Machine learning techniques can be used in Instance Matching to determine if two instances of two different ontologies correspond or not to the same entity, i.e., if they match. In order to do that, the machine learning system learns, through a set of examples called training set, how to distinguish between matching or non-matching instances. The training set works as empirical data that teaches the system how to predict the matches on input data, i.e., the test set.

The training set is a set of examples which contains many correct matches (positive examples) and incorrect matches (negative examples), that is used to train a classifier, resulting in the creation of a model. The model classifies/predicts each example of the test set as belonging to the categories of *yes* or *no* match. Furthermore, the model assigns to both of its predictions a probability distribution, that works as a confidence score for its decisions. In the *yes* category, higher is the probability distribution stronger is the prediction; in the *no* category, lower is the probability distribution stronger is the prediction.

The examples of either training and test sets are organized as a set of features/attributes that are used by the machine learning system as a criteria where it can learn (in the training set) and predict (in the test set). The set of attributes, defined by the developer, can be composed of: the instances' labels and their length; the Jaccard Similarity Coefficient between the labels; a Boolean value representing if both first and last names of two instances' labels are equal; etc.

In order to implement the machine learning matcher, the Weka software (Hall et al., 2009) provides the necessary packages to do it. Here, the training and test sets are represented in ARFF (attribute-relation format file) format.

```
@RELATION training_set_example

@ATTRIBUTE instanceA STRING
@ATTRIBUTE instanceALabel STRING
@ATTRIBUTE instanceALabelLen NUMERIC
@ATTRIBUTE instanceB STRING
@ATTRIBUTE instanceBLabel STRING
@ATTRIBUTE instanceBLabelLen NUMERIC
@ATTRIBUTE firstLastSame {true,false}
@ATTRIBUTE JCValue NUMERIC
@ATTRIBUTE match? {yes,no}

@DATA

"http://instanceA/durao_barroso","durao barroso",13,"http://instanceB/durao_barroso","durao barroso",13,true,1.0,yes
"http://instanceA/durao_barroso","durao barroso",13,"http://instanceB/antonio_guterres","antonio guterres",16,false,0.0,no
"http://instanceA/jorge_coelho","jorge coelho",12,"http://instanceB/jorge_coelho","jorge coelho",12,true,1.0,yes
"http://instanceA/jorge_coelho","jorge coelho",12,"http://instanceB/antonio_seguro","antonio seguro",14,false,0.0,no
```

Figure 2.11: Training set .ARFF example

```
@RELATION test_set_example

@ATTRIBUTE instanceA STRING
@ATTRIBUTE instanceALabel STRING
@ATTRIBUTE instanceALabelLen NUMERIC
@ATTRIBUTE instanceB STRING
@ATTRIBUTE instanceBLabel STRING
@ATTRIBUTE instanceBLabelLen NUMERIC
@ATTRIBUTE firstLastSame {true,false}
@ATTRIBUTE JCValue NUMERIC
@ATTRIBUTE match? {yes,no}

@DATA

"http://instanceA/paulo_portas","paulo portas",12,"http://instanceB/paulo_portas","paulo portas",12,true,1.0,?
"http://instanceA/paulo_portas","paulo portas",12,"http://instanceB/carlos_martins_portas","carlos martins portas",21,false,0.4621,?
"http://instanceA/paulo_portas","paulo portas",12,"http://instanceB/paulo_sacadura_cabral_portas","paulo sacadura cabral portas",28,true,0.4513,?
"http://instanceA/paulo_portas","paulo portas",12,"http://instanceB/joaquim_jose_nunes_portas","joaquim jose nunes portas",25,false,0.4155,?
"http://instanceA/paulo_portas","paulo portas",12,"http://instanceB/joaquim_j_nunes_portas","joaquim j nunes portas",22,false,0.3488,?
```

Figure 2.12: Test set .ARFF example

```
"http://instanceA/paulo_portas","paulo portas",12,"http://instanceB/paulo_portas","paulo portas",12,true,1.0,yes,1.0
"http://instanceA/paulo_portas","paulo portas",12,"http://instanceB/carlos_martins_portas","carlos martins portas",21,false,0.4621,no,0.0
"http://instanceA/paulo_portas","paulo portas",12,"http://instanceB/paulo_sacadura_cabral_portas","paulo sacadura cabral portas",28,true,0.4513,yes,1.0
"http://instanceA/paulo_portas","paulo portas",12,"http://instanceB/joaquim_jose_nunes_portas","joaquim jose nunes portas",25,false,0.4155,no,0.0
"http://instanceA/paulo_portas","paulo portas",12,"http://instanceB/joaquim_j_nunes_portas","joaquim j nunes portas",22,false,0.3488,no,0.0
```

Figure 2.13: Predictions example

The Figures 2.11 and 2.12 represent examples of training and test sets in the .ARFF format. The .ARFF files have two distinct sections: the **Header** section, which is composed of the name of the relation, a set of attributes and their types (**STRING**, **NUMERIC**, **nominal-specification** ...); the **Data** section, where each line corresponds to a data to predict and each column, separated by comma, corresponds to the respective attribute.

In these Figures the set of attributes has the following elements: the pair of instances (*instanceA* and *instanceB*); the instances' labels (*instanceALabel* and *instanceBLabel*); their length (*instanceALabelLen* and *instanceBLabelLen*); a Boolean value representing if both first and last names of two instances' labels are equal (*firstLastSame*); the Jaccard Similarity Coefficient between the labels (*JCValue*). There is also the *match?* attribute where it is indicated if the pair of instances match (*yes*) or not (*no*). In the training set, the attribute is already assigned (*yes* - positive examples or *no* - negative examples) in the **Data** section, in order to teach the system how to distinguish between matching or non-matching instances. And in the test set, the same attribute as a question mark in the **Data** section, where the system places its prediction (*yes* or *no*). Note: in both sets, the **STRING** type attributes are merely for human information purposes. They are not considered in the processes of training and prediction.

The Figure 2.13 represents the system's prediction, and the question marks originally from the test set (Figure 2.12), were replaced by *yes* or *no*. According to the system's prediction, the pair of instances ("http://instanceA/paulo_portas","http://instanceB/paulo_portas") and ("http://instanceA/paulo_portas","http://instanceB/paulo_sacadura_cabral_portas") match, i.e., they correspond to the same entity. The last value of each line represents the probability distribution of each prediction.

2.3.1 Set of Attributes

As said before, both training and test sets have a set of attributes, defined by the developer, that gives to the machine learning system a set of criteria where it can learn (in the training set) and predict (in the test set). This dissertation used attributes for its instance matching work, that were already proposed in other works. The attributes based on the paper (Rong et al., 2012): *double idfSim1; double topIdfSim2; double idfSim3; double topIdfSim4; double cosSim5; double idfSim6; double topIdfSim7; double editSim8; double countSim9; double countSim10; double countSim11*; and the attributes proposed by the REACTION group: *int name1Len; int name2Len; boolean firstSame; double firstSameEC; boolean lastSame; double lastSameEC; boolean twoLastSame; double twoLastSameEC; boolean firstLastSame; double firstLastSameEC; double jcValue*.

There are some considerations to be made about the attributes proposed by the REACTION group. In this dissertation, they were only used for instance's labels.

$$WordEC(w) = -\log(\frac{Freq(w)}{MaxFreq}).$$

Figure 2.14: The evidence content formula for a word
Source: (Couto et al., 2005)

The evidence content formula works as follows: the less frequent is a word within a domain, higher is its evidence content. The idea is: if the instance A_1 from the Ontology 1 and the instance B_1 from the Ontology 2 have in their labels the same uncommon word (low frequency), they are more likely to represent the same entity, i.e., to match.

In this work, the domain corresponds to all instances belonging to the ontologies under process, where the frequency of a word ($Freq(w)$) is just counted once per instance. The $MaxFreq$ corresponds to the maximum frequency found. A word that is just found once has high evidence content, and a word that corresponds to the maximum frequency has no evidence content.

$$NameEC(n) = \sum_{w \in Words(n)} WordEC(w)$$

Figure 2.15: The evidence content formula for a name. The evidence content of a name is the sum of the evidence content of its words.
Source: (Couto et al., 2005)

Considering that a name is composed of a set of words ($Words(n)$), its evidence content ($NameEC(n)$) is calculated through the sum of the evidence content of them ($\sum WordEC(w)$).

int nameXLen The length of the label of an instance. For example, instance A_1 with the label: "blue"; **nameXLen** = 4. The *name1Len* and *name2Len* attributes represent respectively the length of the labels of the pair of instances under process;

boolean firstSame Whether or not two instances have in their labels equal first names. For example, the instance A_1 from the Ontology 1 with the label: "**jose** manuel durao barroso"; and the instance B_1 from the Ontology 2 with the label: "**jose** durao barroso". **firstSame** = *true*. Moreover, for cases where the labels have just one name, the attribute works as well. For example, the instance A_2 from the Ontology 1 with the label: "**barroso**"; and the instance B_2 from the Ontology 2 with the label: "**barroso**". **firstSame** = *true*;

double firstSameEC It returns the **e**vidence **c**ontent of the word (Figure 2.14) that represents the equal first name between two instances' labels. If the first name is not the same, i.e., **boolean firstSame** = *false*, it returns 0.0;

boolean lastSame Whether or not two instances have in their labels equal last names. For example, the instance A_1 from the Ontology 1 with the label: "jose manuel durao **barroso**"; and the instance B_1 from the Ontology 2 with the label: "jose durao **barroso**". **lastSame** = *true*. Moreover, for cases where the labels have just one name, the attribute works as well. For example, the instance A_2 from the Ontology 1 with the label: "**barroso**"; and the instance B_2 from the Ontology 2 with the label: "**barroso**". **lastSame** = *true*;

double lastSameEC It returns the **e**vidence **c**ontent of the word (Figure 2.14) that represents the equal last name between two instances' labels. If the last name is not the same, i.e., **boolean lastSame** = *false*, it returns 0.0;

boolean twoLastSame Whether or not two instances have in their labels equal two last names. For example, the instance A_1 from the Ontology 1 with the label: "jose manuel **durao barroso**"; and the instance B_1 from the Ontology 2 with the label: "jose **durao barroso**". **twoLastSame** = *true*. **But**, for cases where the labels have just one name, the attribute **does not** work (it returns *false*). For example, the instance A_2 from the Ontology 1 with the label: "**barroso**"; and the instance B_2 from the Ontology 2 with the label: "**barroso**". **twoLastSame** = ***false***;

double twoLastSameEC It returns the **e**vidence **c**ontent of the name (Figure 2.15) that represents the equal two last names between two instances' labels. If the two last names are not the same, i.e., **boolean twoLastSame** = *false*, it returns 0.0;

boolean firstLastSame Whether or not two instances have in their labels equal first and last names. For example, the instance A_1 from the Ontology 1 with the label: "**jose** manuel durao **barroso**"; and the instance B_1 from the Ontology 2 with the label: "**jose** durao **barroso**". **firstLastSame** = *true*. Moreover, for cases where the labels have just one name, the attribute works as well. For example, the instance A_2 from the Ontology 1 with the label: "**barroso**"; and the instance B_2 from the Ontology 2 with the label: "**barroso**". **firstLastSame** = *true*;

double firstLastSameEC It returns the **e**vidence **c**ontent of the name (Figure 2.15) that represents the equal first and last names between two instances' labels. If the first and last names are not the same, i.e., **boolean firstLastSame** = *false*, it returns 0.0;

double jcValue This attributes is related with the Jaccard Similarity Coefficient, already presented in the section: Element-level matching Techniques.

There are also some considerations to be made about the attributes based on the paper (Rong et al., 2012).

$$\mathrm{idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

Figure 2.16: The inverse document frequency formula
Source: `http://en.wikipedia.org/wiki/Tf-idf`

The **i**nverse **d**ocument **f**requency is a measure of whether the term is common or rare across all documents. In this work, the total number of documents ($|D|$) corresponds to the: $\overline{Instances(ontology\,1) + Instances(ontology\,2)}$; each document d correspond to each instance, and each term/word is just counted once per instance. Lower is the number of instances where the word appears, higher is the IDF for that word. The idea is: if the instance A_1 from the Ontology 1 and the instance B_1 from the Ontology 2 have in their textual information the same uncommon word (low frequency), they are more likely to represent the same entity, i.e, to match.

double topIdfSim The *topIdfSim2*, *topIdfSim4* and *topIdfSim7* attributes are calculated based on the same formula. Given two sets of words T_1 and T_2, it is extracted respectively from them, the words with the highest IDF values (topIdf): W_1 and W_2. Then, the IDF values of the words from W_1 that intersect the words from the T_2, and the IDF values of the words from W_2 that intersect the words from the T_1, are all summed. This sum is then divided by the value that results from the sum of the W_1 and W_2 IDF values;

double idfSim The *idfSim1*, *idfSim3* and *idfSim6* attributes are calculated based on the same formula. This formula is the same of the **double topIdfSim**, but with the difference that the words with the highest IDF values (topIdf) are not extracted. Thus, the W_1 and W_2 represent all the words that make part of the sets of words T_1 and T_2 respectively, and not just the ones with the highest IDF values;

double cosSim The *double cosSim5* attribute is calculated as follows: given two sets of words T_1 and T_2 is computed their cosine similarity[5];

double editSim The *double editSim8* attribute is calculated through a formula that uses the edit distance principle. Given two sets of words T_1 and T_2, it is measured the minimum number of single-character edits necessary to make the T_1 and T_2 the same. These edits can be the insertion of a character, the deletion of a character, or the substitution of a character in either T_1 or T_2. The edit distance algorithm used was the Levenshtein distance (Levenshtein, 1966);

[5]http://en.wikipedia.org/wiki/Cosine_similarity

double countSim The *countSim9*, *countSim10* and *countSim11* attributes are calculated through a formula that counts the number of common words between two sets of words T_1 and T_2.

2.3.2 Cross-Validation technique

Cross-Validation is a technique for estimating the performance of a predictive model. Here, the training and test sets are from the same dataset. In machine learning approaches is common to use the 10-fold cross-validation, where the dataset is divided into 10 equally (or nearly equally) sized parts/folds. Then, 10 iterations are performed such that within each iteration a different fold is used to validate the model (the fold is used as test set), being the other 9 folds (the training set), along with a classifier, used to generate the model. Furthermore, in the **stratified** 10-fold cross-validation, the dataset is stratified to make sure that each of the 10 folds contains roughly the same proportions of the two categories (in the context of this work: *yes* - correct matches and *no* - incorrect matches). Note: when a fold is used as test set, the already known categories are not considered. Once again, the Weka software provides the necessary packages to implement this technique.

2.3.3 Classifiers

The Weka software already provides several classifiers. The classifiers used in this work are presented in the list below. The criteria to choose them was to encompass a variety of groups of classifiers. The groups are: **Trees**; **Meta**; **Bayes**; **Functions**; and **Neural Networks**.

Trees

J48 (Quinlan, 1993) is an open source Java implementation of the C4.5 algorithm in the Weka data mining tool;

Random forests are an ensemble learning method for classification (and regression) that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees[6]. The algorithm was developed by Leo Breiman (Breiman, 2001) and Adele Cutler;

Random Tree constructs a tree that considers K randomly chosen attributes at each node[7].

Meta

Rotation forest (Rodriguez et al., 2006) every decision tree is trained by first applying principal component analysis (PCA) on a random subset of the input features.

[6] http://en.wikipedia.org/wiki/Random_forest

[7] http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/RandomTree.html

Bayes

NaiveBayes classifier (John and Langley, 1995) is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions[8].

Functions

SMO (Platt, 1998) implements John Platt's sequential minimal optimization algorithm for training a support vector classifier[9].

Neural Networks

Multilayer Perceptron (Rosenblatt, 1961) utilizes a supervised learning technique called back propagation for training the network.

2.4 Performance Evaluator

The Precision, Recall and Accuracy metrics are used to assess the quality of the alignments produced by the instance matching techniques, and they are calculated when the comparison between the alignment and the reference alignment is made. The alignment can be one-to-one or many-to-many. In the first case, each instance is only present once in the alignment, i.e., there are no more than one match per instance; in the second case, each instance can be present several times in the alignment, i.e., it can be more than one match per instance. Furthermore, there is also the F-measure metric that is the harmonic mean of Precision and Recall.

In order to calculate these metrics it is necessary to make the further calculation:

	actual class (expectation)	
predicted class (observation)	**tp** (true positive) Correct result	**fp** (false positive) Unexpected result
	fn (false negative) Missing result	**tn** (true negative) Correct absence of result

Figure 2.17: Classification table
Source:http://en.wikipedia.org/wiki/Precision_and_recall

Actual class (expectation) = Reference alignment
Predicted class (observation) = Alignment

Instances (ontology a) = $\{a_1,...,a_n\}$
Instances (ontology b) = $\{b_1,...,b_m\}$

[8]http://en.wikipedia.org/wiki/Naive_Bayes_classifier
[9]http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html

Reference alignment = $\{(a_i,b_j),...\}$ 1<= i <=n and 1<= j <=m
Alignment = $\{(a_k,b_l),...\}$ 1<= k <=n and 1 <= l <=m
Total domain (one-to-one) = $\overline{reference\ alignment}$ + $\overline{Instances(ontology\ a) - reference\ alignment}$
+ $\overline{Instances(ontology\ b) - reference\ alignment}$
Total domain (many-to-many) = $\overline{Instances(ontology\ a) * Instances(ontology\ b)}$

TP = $\overline{\{(a_x, b_y)\ :\ (a_x, b_y)\ in\ alignment\ and\ (a_x, b_y)\ in\ reference\ alignment\}}$
= $\overline{\{alignment\ \cap\ reference\ alignment\}}$

FP = $\overline{\{(a_x, b_y)\ :\ (a_x, b_y)\ in\ alignment\ and\ (a_x, b_y)\ not\ in\ reference\ alignment\}}$
= $\overline{\{alignment\ -\ TP\}}$

FN = $\overline{\{(a_x, b_y)\ :\ (a_x, b_y)\ not\ in\ alignment\ and\ (a_x, b_y)\ in\ reference\ alignment\}}$
= $\overline{\{reference\ alignment\ -\ TP\}}$

TN = $\overline{\{(a_x, b_y)\ :\ (a_x, b_y)\ not\ in\ alignment\ and\ (a_x, b_y)\ not\ in\ reference\ alignment\}}$
= $\overline{\{total\ domain\ -\ \{alignment\ +\ \{reference\ alignment\ -\ TP\}\}\}}$

Precision: is the fraction of correct matches in the alignment. The correct matches are the matches present in the reference alignment.

$$\text{Precision} = \frac{tp}{tp + fp}$$

Figure 2.18: Precision formula
Source:http://en.wikipedia.org/wiki/Precision_and_recall

Recall: is the fraction of matches of the reference alignment that are present in the alignment.

$$\text{Recall} = \frac{tp}{tp + fn}$$

Figure 2.19: Recall formula
Source:http://en.wikipedia.org/wiki/Precision_and_recall

F-measure: is the average of Precision and Recall rates.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Figure 2.20: F-measure formula
Source:http://en.wikipedia.org/wiki/F-measure

Accuracy: is the fraction of true results over all the matching possibilities (total domain) between the ontologies aligned.

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

Figure 2.21: Accuracy formula
Source:http://en.wikipedia.org/wiki/Precision_and_recall

2.5 PHP built-in Functions

PHP[10] is a server-side scripting language designed mainly for web development, which provides several built-in functions.

1. **trim** function — Strip white space (or other characters) from the beginning and end of a string.
2. **strip_tags** function — Strip HTML and PHP tags from a string.
3. **htmlentities** function — Convert all applicable characters to HTML entities; flags: ENT_QUOTES - Will convert both double and single quotes; encoding: UTF-8 - ASCII compatible multi-byte 8-bit Unicode.
4. **urlencode** function — URL-encodes string. This function is convenient when encoding a string to be used in a query part of a URL, as a convenient way to pass variables to the next page.

[10]http://php.net/

Chapter 3

Implementation

This chapter describes the architecture of the system developed and its modules. The system's main goal was solving OAEI 2012 and POWER instance matching problems, as any other instance sets following the same formats.

3.1 System Architecture

This section presents the input and output of the system, its modules, their relations, and the data that flows between them.

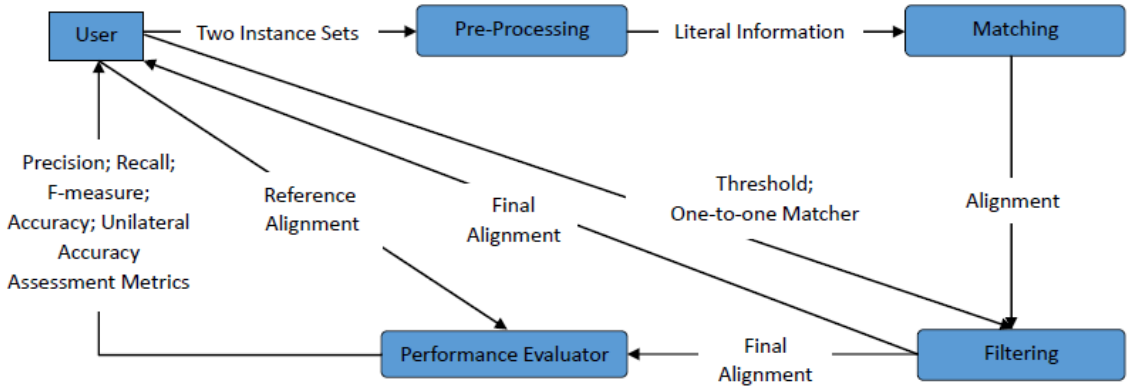


Figure 3.1: Overview of the System. This Figure shows the modules that are part of the system, the data flow between them, the user input and what it receives as output.
Based on: (Larman, 2004)

The system receives as input two instance sets introduced by the user of the system, and produces to the output the final alignment. The final alignment is composed of: instances for which was identified a match, i.e., the pairs of instances matched; the instances to which no match was identified, i.e., the instances matching to NULL; and their confidence score. The confidence score [0,1] represents the probability of each match to be true. Optionally, the user can also input a reference alignment. This allows the system to output metrics that attest the quality of the final alignment, by comparing it with the

reference alignment. It is assumed that the reference alignment contains all the correct matches between the two instance sets.

In order to do that, the system on receiving as input the two instances sets, performs a **pre-processing** operation, in order to extract the data of each instance, i.e., the literal information. This is fundamental to perform the instance **matching** process because, it provides to the matcher the necessary information to determine if two instances correspond to the same entity or not, i.e., if they match. The outcome of it is the alignment. Furthermore, before becoming the final alignment, is applied to it a **filtering** process. This process receives two parameters given by the user on input. The first parameter is the Threshold and allows the user to select the minimum confidence score (0.0 by default) of the matches to be included in the final alignment; and the second parameter is the one-to-one matcher, which is optional, and guarantees that in the final alignment, each instance is only present once in the final alignment, i.e., there are no more than one match per instance. If a reference alignment is given as well, the final alignment is assessed in its **performance**, through the metrics of Precision, Recall, F-measure, Accuracy and Unilateral Accuracy.

3.2 Pre-Processing

This module receives as input two instance sets, introduced by the user of the system, and through the Apache Jena[1], extracts the literal information belonging to each instance.

The literal information of an instance is the textual content of its properties, and can be composed of: labels, dates, articles, codes, etc. For example, the following instance representation:

```
<instance:Instance rdf:about="http://tool.for.ontology.instance.matching/jose_manuel_durao_barroso">
<instance:name>José Manuel Durão Barroso</instance:name>
<instance:date_of_birth>23/03/1956</instance:date_of_birth>
<instance:article>José Manuel Durão Barroso é um político e professor português,
actual presidente da Comissão Europeia, cargo que ocupa desde Novembro de 2004.
Em Portugal, foi sub-secretário do ministério dos assuntos internos, em 1985,
e ministro dos Negócios Estrangeiros em 1992.
Entre 2002 e 2004, ocupou o cargo de primeiro-ministro da República Portuguesa.
</instance:article>
</instance:Instance>
```

Above, is shown an instance, represented in .RDF, corresponding to the Portuguese politician Durão Barroso. The first line, is the URI of the instance. The following lines correspond to the properties of the instance: name (label), date of birth, and article. The content of the properties is the literal information: José Manuel Durão Barroso, 23/03/1956, José Manuel Durão Barroso é um político e professor português...

This pre-processing operation returns two objects, corresponding to each instance set introduced, that maps each instance to the correspondent literal information. Because, this system only deals with the labels of the instances, in order to be able to perform instance matching between all kinds of instance sets, only the literal information corresponding to

[1] http://jena.apache.org/

the labels is extracted. This pre-processing operation was applied in the POWER, DBpediaPT and Verbetes instance sets.

However, for the OAEI 2012 matching problem the pre-processing operation had to be different. The literal information extracted corresponded to the properties such as: name and amount (both considered as labels); article; date_of_birth; form_of_government; capital; etc. Furthermore, it was also necessary to recur to the SPARQL provided by the Apache Jena to identify the instances, because this international contest introduced transformations at logical level that made harder to identify them.

3.2.1 Pre-Processing Sub-Modules

In the sub-sections below are described the pre-processing sub-modules, that are responsible for cleaning the literal information of the instances. This is done, with the purpose of putting all of them in the same textual conditions, in order to enhance the matchers performance.

Diacritical Eliminator

This sub-module is responsible for eliminating the diacritics that might exist in the literal information of an instance. For example, an instance with the label: "José Manuel Durão Barroso" is transformed to "Jose Manuel Durao Barroso", by removing the accents. It was applied in POWER, DBpediaPT and Verbetes, because their labels were in Portuguese.

Literal Information Normalizer

This sub-module is responsible for performing a deeper cleaning operation on the literal information of the instances. Because, this system only deals with the labels of the instances, the cleaning process consists of: putting the labels to lower case; remove all contents placed within parentheses; remove all non-word characters; replace all underscore characters for a single white space; replace all multiple white spaces for a single one; remove the characters that are duplicate in row in each word; and finally, an operation of trim is executed. This process was applied in the POWER, DBpediaPT and Verbetes instance sets.

However, for the OAEI 2012 matching problem had to be created a different literal information normalizer. Because, it was introduced strong textual changes in the literal information of the properties. These changes were generally random text and words modifications. To face and overcome it, the cleaning process consisted of: putting the literal information to lower case; remove all contents placed within parentheses; remove all non-word characters; replace all underscore characters for a single white space; replace all multiple white spaces for a single one; separate numbers from characters in a word; remove the characters that are duplicate in row in each word; remove single characters; remove the words that do not have vowels nor numbers; and finally, an operation of trim is executed. This process of normalization was not perfect and could sometimes produce errors by eliminating some important information. But, the harmful text that it cleaned paid off.

3.3 Matching

This module is responsible for performing instance matching between the instance sets introduced (already pre-processed), and to return the alignment. Basically, the instance matching mechanism tries to match each instance of one instance set with each instance of the other instance set.

3.3.1 Element-level matcher

This sub-module implements the FirstLastNamePlusJaccard matcher. This matcher, is an algorithm that considers a match the instances that have in their labels equal first and last names. The confidence score of each match is set by the Jaccard Similarity Coefficient (Jaccard, 1912) between the two labels. If the instances have multiple labels, each pair of labels are evaluated in their first and last names. In case of draw, i.e., in cases that there are more than one pair of labels that matches, the pair of labels that wins, is the one that has the highest confidence score set by the Jaccard Similarity Coefficient.

For example, assume that Instance set 1 has an instance with the labels: "jose manuel durao baroso" and "durao baroso"; and Instance set 2 has an instance with the labels: "jose durao baroso" and "durao baroso". The matcher will match these two instances, because the pairs of labels ("**jose** manuel durao **baroso**","**jose** durao **baroso**") and ("**durao baroso**","**durao baroso**"), have the same first and last names. The confidence score of the match is set by the pair ("durao baroso","durao baroso",**1.0**), because it is higher than the other pair ("jose manuel durao baroso", "jose durao baroso",**0.75**). Moreover, for cases where the labels have just one name the matcher works as well. Assume that Instance set 1 has an instance with the label: "**durao**"; and Instance set 2 has an instance with the label: "**durao**"; the matcher will consider that these two instances have the same first and last names. This matcher was used as baseline in the results presented in the Results Chapter 4.

3.3.2 Machine Learning matcher

This sub-module implements the machine learning matcher, through the Weka software (Hall et al., 2009). To decide which pairs of instances are to be matched or not, this matcher uses a model, whose training set was created from the POWER instance set against the DBpediaPT instance set, and classified by the Rotation Forest. This classifier was chosen because it proved to be one of the top classifiers during the OAEI 2012 tests, as explained in the Results Chapter 4. This matcher only deals with the labels of the instances, in order to be able to perform instance matching between all kinds of instance sets. For that reason, the machine learning attributes that it uses, are based on the labels of the instances. These attributes are: *boolean firstSame; double firstSameEC; boolean lastSame; double lastSameEC; boolean twoLastSame; double twoLastSameEC; boolean firstLastSame; double firstLastSameEC; double jcValue; double editSim; double countSim*. The last two attributes are based on the paper (Rong et al., 2012). Note: the *int name1Len* and the *int name2Len* attributes, are not present in the set, because they were never selected during the attribute selection exercise, that is presented in the Results Chapter 4. Because the instance sets can have several labels for each instance, it is pos-

sible to use them as synonyms to improve the matching possibilities. The pair of labels that produces the highest non-boolean attributes sum, chooses the related attribute values (including the boolean attributes) that will represent the pair of instances in the matching process.

This matcher was used in the POWER-Verbetes and in the POWER-POWER alignment, that are mentioned in the Results Chapter 4.

3.3.3 Instances matching to NULL

In both matchers, they assign a confidence score to the instances matching to NULL. An instance that matches to NULL, is an instance that was not matched to any other instance of the other instance set. However, the matchers assign a score for each match, even if it is a *no* match. The confidence score of an instance matching to NULL is set by the formula: $1 - (MaxScore)$. Where the second term of the formula, is the maximum score, among the *no* matches, assigned by the matcher for that instance. Then, it is subtracted by one, because the higher a confidence score is, the lowest is the probability of not having a matching instance in the other set.

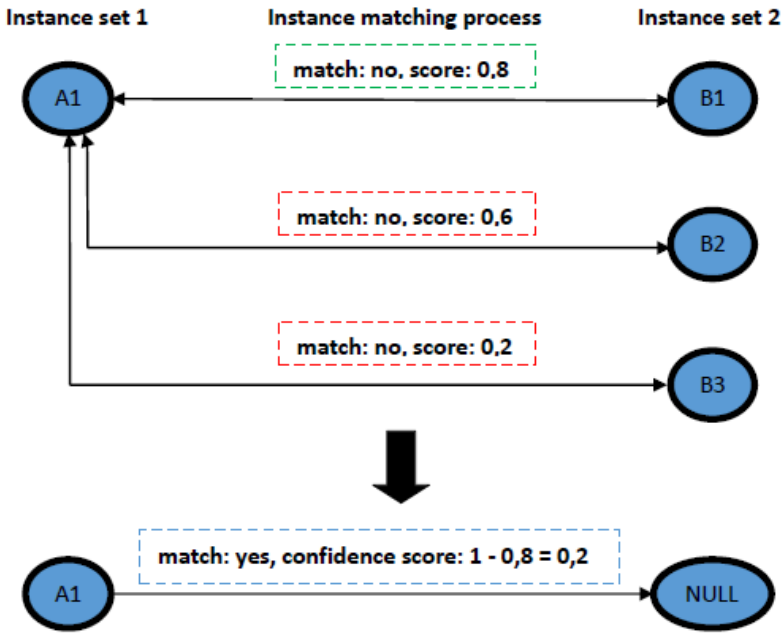


Figure 3.2: NULL's confidence score setting by the instance matching process.

The Figure 3.2 shows how the confidence score of an instance matching to NULL is set by the instance matching process. The instance A_1 of the Instance set 1, matches to NULL because, the matcher did not match it to no-one (*match:no* to all instances of the Instance set 2). But, it assigned a score to each instance matching assessment (*score: X*). The maximum score is chosen (green box - score: 0,8), and the other ones are rejected (red boxes). At the end, the instance A_1 is matched to NULL, and the confidence score is assigned (blue box - confidence score: 1 - 0,8 = 0,2).

3.4 Filtering

This module implements a set of filters on the alignment returned by the matching process. The set of filters is composed by two parameters given by the user. This module is also responsible for outputting the final alignment.

Threshold is the first parameter, and allows the user to select the minimum confidence score [0,1] of the matches to be included in the final alignment. Only the pairs of instances matched and the instances matching to NULL, that have their confidence score within the threshold are going to be present in the final alignment. All the results presented in the Results Chapter 4, were produced within a threshold of 0.0.

The second parameter is the one-to-one matcher, which is optional, and guarantees that each instance is only present once in the final alignment, i.e., there are no more than one match per instance. This can be done by using the bipartite graph matching approaches. This approach divides the pair of instances matched, into two disjoint sets, being each set composed by the instances of the same instance set. In this case, it matches one instance of one disjoint set to only one instance of the other disjoint set, guaranteeing that the confidence score of the match is the highest possible for the instances composing the pair. This is not valid for the (A) labels-sum+alphabetic-order one-to-one matcher, whose algorithm is explained below. Note: the instances matching to NULL are not considered in this process, because they do not pair to any instance of the other disjoint set. This module incorporates three one-to-one matchers, each one having their own criteria to choose the instances that should pair. The matchers are: (A) labels-sum+alphabetic-order; (B) confidence-score+attribute-sum; Hungarian Algorithm (Kuhn, 1955), that is an algorithm for constructing a maximum weight perfect matching in a bipartite graph.

The (A) one-to-one matcher is based on two criteria. For example, given the instances A_1 with the label "a1", A_2 with the label "a2", and A_3 with the label "a3", from the Instance set 1, that are matched with the instance B_1 with the label "b1", from the Instance set 2. The first criteria, chooses the pair whose the sum of their labels length is the highest; if there is a draw (in the example there is, because "a1b1", "a2b1", "a3b1" length is four), the second criteria chooses the pair, within the tied pairs, that is in the first place of the alphabetic order. To assess that, the labels of the pair of instances are concatenated. If there is still a draw, the pair chosen is randomly selected among the pairs that are placed in first. Following the example, the pair (A_1,B_1) is chosen, this means that the instances A_2 and A_3 are excluded. In this scenario, they are matched to NULL because they do not match to any other instance. It is also assigned a new confidence score for these two instances. Once again, they are going to be present in the final alignment, only if their (new) confidence scores are within the threshold.

As the previous matcher, (B) one-to-one matcher is based on two criteria as well. The first criteria is the highest confidence score, and the second criteria is the highest attribute sum. The attribute sum for each pair is set by the matchers. In the machine learning matcher, it corresponds to the sum of the non-boolean attributes. In the FirstLastNamePlusJaccard matcher, it corresponds to the *double editSim* and *double countSim* sum.

If the one-to-one matcher parameter is not chosen, matches of one instance from one disjoint set to multiple instances of the other disjoint set (many-to-many matches), may appear in the final alignment. In practice, the alignment will just be transformed into the final alignment (after the Threshold selection).

3.4.1 Instances matching to NULL

As said before, in the one-to-one matcher there can be instances when are excluded match to NULL, because they do not match to any other instance. In this case, a new confidence score is assigned to these instances. The formula is $1 - (MaxScore)$, which was already introduced in a previous section. Following the example above, the second term of the formula is the confidence score of the match of the instance excluded.

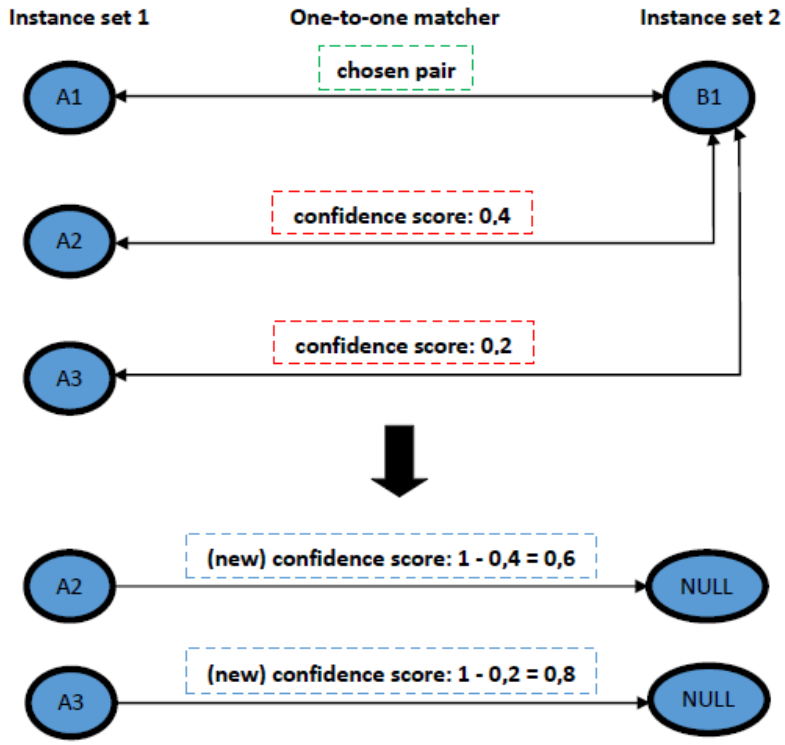


Figure 3.3: NULL's confidence score setting in the one-to-one matcher - scenario 1

The Figure 3.3 shows how the new confidence scores of the instances of the example are set. The instances A_2 and A_3 of the Instance set 1 were excluded of the match with the instance B_1 of the Instance set 2. These instances are then matched to NULL, being their respective confidence scores (red boxes - *confidence score: X*) subtracted by one, assigning to these instances new confidence scores (blue boxes - *(new) confidence score: 1 - X = Y*).

Furthermore, there can be situations where an instance is excluded from several matches. In this case, the second term of the formula is the maximum confidence score among the matches of the instance excluded.

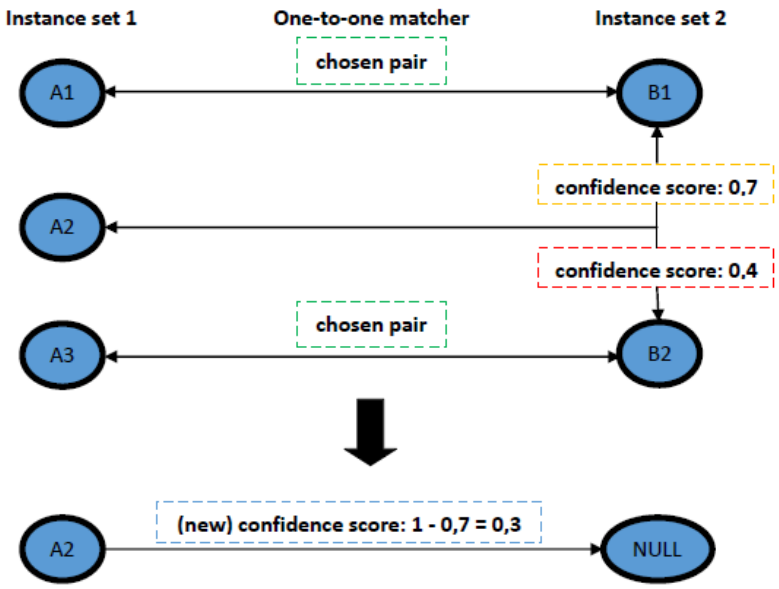


Figure 3.4: NULL's confidence score setting in the one-to-one matcher - scenario 2

The Figure 3.4 shows how the confidence score of an instance that was excluded by the one-to-one matcher is set. The instance A_2 of the Instance set 1 matches to NULL because, the one-to-one matcher decided to match the instances B_1 and B_2 from the Instance set 2, to other instances from the Instance set 1 (green boxes - *chosen pair*). The maximum confidence score among the matches of the instance A_2 is chosen (yellow box - confidence score: 0,7), and the instance is matched to NULL with a new confidence score assigned (blue box - (new) confidence score: 1 - 0,7 = 0,3).

3.5 Performance Evaluator

This module is responsible for assessing the quality of the final alignment returned by the filtering module. In case the user gives as input a reference alignment, which contains all the correct matches between the instance sets, it is calculated and produced to the output the following performance metrics: Precision; Recall; F-measure; and Accuracy.

Furthermore, this module also calculates the Unilateral Accuracy metric for each instance set. This metric shows the number of instances of each instance set, that are correctly matched over the total of instances composing the respective instance set. In this context, correctly means the instances that are matched in the final alignment that are present in the reference alignment; and the instances matching to NULL in the final alignment that are not present in the reference alignment.

In the sections below, it is presented the part of this work where it was implemented the cross-validation technique. This part was used to evaluate the cross-validation matcher performance and therefore to produce metrics.

Cross-validation technique

In the matching problems below, it was used the machine learning technique of stratified 10-fold cross-validation. The dataset, where the training and test sets are extracted from, is created through the association of each instance of one instance set with each instance of the other instance set. In each association, the set of attributes is extracted and the *yes* (match) and *no* (do not match) categories are assigned based on the reference alignment between the instance sets. For example, on associating the instance A_1 of the instance set 1 with the instance B_1 of the instance set 2, if this pair of instances is present in the reference alignment, it is assigned the *yes* category.

In the following matching problems, the datasets are built from the instance sets that are to be (re) aligned. This happened because was not used other instance sets where the training set could have been extracted from. The reasons are explained below. The training set is used to train the classifier, resulting in the creation of the (instance matching) predictive model.

3.5.1 OAEI 2012 instance matching

In the OAEI 2012 matching problem, the usage of stratified 10-fold cross-validation machine learning technique is justified, by the absence of other instance sets (outside the OAEI 2012) capable of providing literal information suitable enough to fulfil the attributes requirements. The set of attributes chosen were, the ones related to the instances' labels: *int name1Len; int name2Len; boolean firstSame; double firstSameEC; boolean lastSame; double lastSameEC; boolean twoLastSame; double twoLastSameEC; boolean firstLastSame; double firstLastSameEC; double jcValue;* and the ones based on the paper (Rong et al., 2012): *double idfSim1; double topIdfSim2; double idfSim3; double topIdfSim4; double cosSim5; double idfSim6; double topIdfSim7; double editSim8; double countSim9; double countSim10; double countSim11*. I sent an e-mail to the authors of this paper to find out, which datasets did they use to build the model in order to perform instance matching, but the reply was inconclusive. This paper proposes an approach where the attributes are build based on the combination of literal information:

Paper Attributes	Combination of Literal Information
idfSim1	l_{single}
topIdfSim2	l_{single}
idfSim3	$l_{single} \cup l_{short} \cup l_{label}$
topIdfSim4	$l_{single} \cup l_{short} \cup l_{label}$
cosSim5	$l_{single} \cup l_{short} \cup l_{label} \cup l_{property} \cup l_{long}$
idfSim6	$l_{single} \cup l_{short} \cup l_{label} \cup l_{property} \cup l_{long}$
topIdfSim7	$l_{single} \cup l_{short} \cup l_{label} \cup l_{property} \cup l_{long}$
editSim8	l_{label}
countSim9	l_{label}
countSim10	l_{date}
countSim11	l_{number}

Table 3.1: Paper Attributes overview. This table refers to the elements of the set of attributes, used in the OAEI 2012 matching problem, that are from the paper (Rong et al., 2012), and the way they are calculated.

The l_{single} literal information is extracted from the instance's properties that have in their literal information only one word; the l_{short} concerns the ones that have between two and three words; the l_{long} concerns the ones that have more than three words; the $l_{property}$ concerns the literal information present in other properties of other instances, that the instances point to; The l_{number} and the l_{date}, concerns respectively to the properties that have in their literal information numbers solely, and dates. It were just considered dates which have only numbers in their representation separated by -, / or :. For example: 24-11-2004; or 11/24/2004. Note: the l_{link} literal information, mentioned in the paper (Rong et al., 2012), was not considered because, in the analysis made on the OAEI 2012 instance sets, no links in the literal information were found. At last, the l_{label} concerned the label of the instances. In the Sandbox and IIMB instance sets (both belonging to the OAEI 2012), the labels of the instances are represented by the *name* or *amount* properties, but there are also instances that do not have labels in their properties.

3.5.2 POWER-DBpediaPT instance matching

To perform instance matching between POWER and DBpediaPT, it was necessary to use the machine learning technique of stratified 10-fold cross-validation (using as classifier the Random Forest), based on the attributes related with the labels of the instances. Because, POWER and DBpediaPT have several labels for each instance, it was possible to use synonyms to improve the matching possibilities. The pair of labels that produced the highest non-boolean attributes sum, chose the related attribute values (including the boolean attributes) to represent the pair of instances in the matching process.

The usage of stratified 10-fold cross-validation is justified by the fact that the model, which training set was created from the Sandbox reference instance set against the Sandbox 001 instance set, both belonging to the OAEI 2012, and classified by the Random Forest, did not produce good metrics in the assessment of the first final alignment. Precision equals to 31.33%, and Recall equals to 58.63%. In order to perform cross-validation and to assess a final alignment, was necessary to have a reference alignment. The one used in this matching problem was built manually by me and provided to the REACTION group[2], and contained instances matching to NULL to fulfil the group demands.

[2]http://dmir.inesc-id.pt/project/Reaction

```
<map>
   <Cell>
    <entity2 rdf:resource="http://pt.dbpedia.org/resource/Luís_Barbosa"/>
    <entity1 rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#luis_eduardo_silva_barbosa_merged"/>
    <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1.0</measure>
    <relation>=</relation>
   </Cell>
  </map>
<map>
   <Cell>
    <entity2 rdf:resource="http://NULL/3115107986776"/>
    <entity1 rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#emanuel_vasconcelos_jardim_fernandes"/>
    <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1.0</measure>
    <relation>=</relation>
   </Cell>
</map>
<map>
   <Cell>
    <entity2 rdf:resource="http://pt.dbpedia.org/resource/José_Dias_Coelho"/>
    <entity1 rdf:resource="http://NULL/2961393280777"/>
    <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1.0</measure>
    <relation>=</relation>
   </Cell>
</map>
```

Figure 3.5: POWER-DBpediaPT reference alignment sample. The URL of the DBpediaPT instances were changed to fulfil the REACTION group demands.

DBpediaPT Filtering

Before performing the instance matching between POWER and DBpediaPT, it was necessary to filter the DBpediaPT instances. This was done because the DBpediaPT was very large (contains 57103 instances) for the instance matching tool to handle. Basically, this filtering process consisted of detecting only instances concerning Portuguese politicians. To detect it, a supporting instance set of the same version of the DBpedia was used[3]. This supporting instance set contains abstract texts for each person of the Portuguese version of Wikipedia, giving several types of information such as the complete name of a person (which was used as a second label for the DBpediaPT instances, in the instance matching process), its nationality, and profession. For example, "José Manuel Durão Barroso é um político e professor português". More precisely, the filter looked for the following patterns: portug; politic; ministr; dirigente; presiden; deputad; partido; autar; vereador; secretari; parlamento; govern; legisla; regiona; assembleia; diplomata. Note that some patterns are incomplete to encompass gender and root words. Were also included as patterns the Portuguese political parties acronyms, but this was not efficient because these patterns let many instances that were not Portuguese politicians to pass.

The outcome of this filtering process was the reduction of the DBpediaPT from 57103 instances to 596. This process was not perfect, because this reduced instance set still had instances that were not Portuguese politicians, and it is possible that some Portuguese politicians instances were not detected.

[3]http://downloads.dbpedia.org/3.8/pt/

<http://pt.wikipedia.org/wiki/Albino_Forjaz_de_Sampaio> <> <> .
<http://pt.wikipedia.org/wiki/Albert_Einstein> <> <> .
<http://pt.wikipedia.org/wiki/Adriano> <> <> .
<http://pt.wikipedia.org/wiki/Afonso,_Príncipe_de_Portugal_(1475-1491)> <> <> .
<http://pt.wikipedia.org/wiki/Alexandre_Rodrigues_Ferreira> <> <> .
<http://pt.wikipedia.org/wiki/Aldous_Huxley> <> <> .
<http://pt.wikipedia.org/wiki/Aleksandr_Oparin> <> <> .
<http://pt.wikipedia.org/wiki/Antônio_Mariano_de_Oliveira> <> <> .
<http://pt.wikipedia.org/wiki/António_Guterres> <> <> .

Figure 3.6: Sample instances from DBpediaPT before the filter process

<http://pt.dbpedia.org/resource/Durão_Barroso> <http://dbpedia.org/ontology/abstract> "**José Manuel Durão Barroso é um político e professor português**, actual presidente da Comissão Europeia, cargo que ocupa desde Novembro de 2004. Em Portugal, foi sub-secretário do ministério dos assuntos internos, em 1985, e ministro dos Negócios Estrangeiros em 1992. Entre 2002 e 2004, ocupou o cargo de primeiro-ministro da República Portuguesa. A 23 de novembro de 2004, Durão Barroso assumiu as funções de Presidente da Comissão Europeia, cargo que irá assumir outra vez em Novembro de 2009, após ter sido reeleito pelo Parlamento Europeu a 16 de Setembro."@pt .

Figure 3.7: Supporting instance. In bold, the complete name, the nationality and the profession, of the person represented in the instance.

<http://pt.dbpedia.org/resource/Abel_Lima_Baptista> <> <> .
<http://pt.dbpedia.org/resource/Abel_Repolho_Correia> <> <> .
<http://pt.dbpedia.org/resource/Abílio_Augusto_Valdez_de_Passos_e_Sousa> <> <> .
<http://pt.dbpedia.org/resource/Acácio_Pereira_Magro> <> <> .
<http://pt.dbpedia.org/resource/Adalberto_Neiva_de_Oliveira> <> <> .
<http://pt.dbpedia.org/resource/Adelaide_Penha_de_Magalhães> <> <> .
<http://pt.dbpedia.org/resource/Adelino_Amaro_da_Costa> <> <> .
<http://pt.dbpedia.org/resource/Adolf_Hitler> <> <> .

Figure 3.8: Sample instances from DBpediaPT after the filter process. In bold, an instance that is not a Portuguese politician. The URL of the instances were changed to fulfil the REACTION group demands.

3.6 Instance Matcher Web Tool

This section describes the Web tool that allows the user to perform instance matching between two instance sets. The Web tool is a front-end for the modules described in the sections above, except for the cross-validation technique module. Furthermore, just one of three one-to-one matchers developed is available as an user option. The one-to-one matcher chosen was the (A) labels-sum+alphabetic-order, because it produced the

best results among the other two: (B) confidence-score+attribute-sum, and Hungarian Algorithm; during the POWER-DBpediaPT alignment, presented in the Results Chapter 4.

The Web tool can be accessed through the URL: `http://lasige.di.fc.ul.pt/webtools/instancematcher/`.

3.6.1 Web Tool Input

The input of the Web tool (Figure 3.10) are: **Instance set 1 (compulsory)** the URI for the instance set to be matched; **Instance label identifiers** the identifiers that indicate which are the instance labels in the instance set 1; **Instance set 2 (compulsory)** the URI for the instance set to be matched; **Instance label identifiers** the identifiers that indicate which are the instance labels in the instance set 2; **Reference alignment** the URI for the reference alignment between the given instance sets; **Instance relation**: One-to-one and many-to-many (selected by default); **Matching algorithm** allows to select the matcher that will perform instance matching between the instance sets given. The matchers are: FirstLastNamePlusJaccard (selected by default) and MachineLearning; **Threshold** allows to select the minimum confidence score [0,1] of the alignment produced to the output; **OAEI 2012** (selected by default) indicates that the instance sets introduced are from the OAEI 2012 contest; **POWER 2010**: Is1 (**I**nstance **s**et **1**) and Is2 (**I**nstance **s**et **2**) indicate respectively which of the instance sets introduced are from the POWER instance set. In both **OAEI 2012** and **POWER 2010** (Is1 and Is2) input, if they are selected, the **Instance label identifiers** input can be left empty.

The **instance label identifiers** are related with the URIs of the properties, within an instance set, that correspond to the labels of the instances. For example:

```
<owl:DatatypeProperty rdf:about="http://oaei.ontologymatching.org/2012/IIMBTBOX/name"/>
<owl:DatatypeProperty rdf:about="http://oaei.ontologymatching.org/2012/IIMBTBOX/amount"/>

<owl:NamedIndividual rdf:about="http://oaei.ontologymatching.org/2012/IIMBDATA/en/abuja">
...
<IIMBTBOX:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Abuja</IIMBTBOX:name>
...
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://oaei.ontologymatching.org/2012/IIMBDATA/m/01xpnvx">
...
<IIMBTBOX:amount rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.8E7</IIMBTBOX:amount>
...
</owl:NamedIndividual>
```

This example, extracted from the Sandbox reference instance set, indicates in the first two lines, the two URIs that identify which are the labels in this instance set. The URIs are: “http://oaei.ontologymatching.org/2012/IIMBTBOX/name”; and “http://oaei.ontologymatching.org/2012/IIMBTBOX/amount”. They identify respectively the labels: <IIMBTBOX:name...>; and <IIMBTBOX:amount...>.

3.6.2 Usage

The input that have the (*) character are mandatory, and their URI must begin with the *http*. If any of them are left empty, an error message is displayed in red color saying: "Missing compulsory input!" (Figure 3.14). To limit the user input, the **Instance set 1, 2** and **Reference alignment** input have a maximum length of 150 characters, and the **Instance label identifiers** have a maximum length of 200 characters. If for any reason these limits have been violated, an error message is displayed in red color saying: "Check the input length: Max = 150 characters! For instance label identifiers: 200 characters!" (Figure 3.15). To add multiple instance label identifiers, is not required any pattern. It is assumed that each identifier begins with the *http* prefix.

The Web tool already provides default input values. In the **Instance set 1**, **2**, **Instance label identifiers** and **Reference alignment**, these values concern the Sandbox instance sets from the OAEI 2012 competition. For experimental purposes these instance sets can be changed from 001 to 011 in the input values (mind the reference alignment input value). If the user changes any input that composes the Web tool, the values will be preserved for the next usage, i.e., do not return to the default values.

The Web tool only supports instance sets from the following extensions: .RDF; .OWL; .NT; and .TTL. On the violation of this requirement, an error message is displayed, in the output page, saying: "Error on Instance Matching! - Problems on the instance matching execution!" (Figure 3.16). If this violation occurs in the **Reference alignment** input, the message will be: "Error on Instance Matching! - Problems on the alignment assessment!" (Figure 3.17). Note that in the .NT and .TTL instance sets extensions no instance label identifiers are needed. This situation also occurs if the **OAEI 2012** and **POWER 2010** input have been selected.

3.6.3 Output

The Web tool output will be in a second page, the output page, through the "Click here to obtain the results!" link, displayed in the input page on submission (Figure 3.10 at the bottom). Until this link does not appear, a loader is displayed near by the submit button "Instance Matching". The output page allows multiple instance matching submissions without loosing the view of the input page. Each user submission has an unique identifier to avoid overwriting. The results appear in the output page by refreshing it, although the page auto-refreshes every 20 seconds. This feature has the advantage of avoiding network time-outs, if the instance matching process takes too long.

The results consist of: the final alignment between the instance sets introduced, available through the link "The alignment" (Figure 3.11); a panel containing information about the input and the elapsed time of the all process. Furthermore, if a reference alignment is given, is also displayed information about the final alignment assessment (Figure 3.12), or if not, just summarized information about it (Figure 3.13).

If some error occurs during the instance matching process, the output page can display the following error messages: "Error on Instance Matching! - Problems on the instance matching execution!" (Figure 3.16), if the error occurs during the instance matching process; "Error on Instance Matching! - Problems on the alignment assessment!" (Figure 3.17), if the error occurs during the alignment assessment process; "Error on Instance Matching! - Problems on the elaboration of instance matching information!", if the error

occurs during the elaboration of the summarized information; or just "Error on Instance Matching!", the default.

3.6.4 Security

Some security issues have already been addressed, such as the maximum input length in characters. But the Web tool has other mechanisms to filter the user input. These mechanisms are provided by the **PHP** language, and are applied before the input is passed as arguments into the system, in the following order: the `trim(input)`; the `strip_tags(input)`; and the `htmlentities(input,ENT_QUOTES,'UTF-8')` operations. Furthermore, is also applied the `urlencode(input)` function to avoid execution of shell or other (malicious) commands. This function is applied as well to the variables that are passed from the input page to the output page. These variables are the unique identifier of the submission, and a flag that signals if a reference alignment was given as input or not. In the output page, these variables are filtered based on the same set and order of the mechanisms described above. Moreover, there is also a maximum character check on these variables. The unique identifier has a maximum length of 30 characters, because during the Web tool tests, it never exceeded fifteen characters; and the reference alignment flag has a maximum length of 1 character, because in the URL variables the *true* value is represented by the number 1, and *false* value is represented by an empty value. On the violation of these requirements the output page will display the following red color error message: "The parameters in the URL must respect a maximum length of: uid <= 30 characters! refAlign_input: <= 1 character!".

3.6.5 Limitations

The Web tool has some limitations that are needed to take into account. The first limitation is about the maximum size of the instance sets introduced as input. If they are too large, the web tool will not return any results, but a red color error message saying: "Error on Instance Matching! - Instance sets too big!" (Figure 3.18), in the output page. The biggest final alignment produced so far by the web tool occurred in the POWER-POWER alignment (2839*2839 = 8.059.921 total instance pairs possibilities). The second limitation is about the type of instances expected. It is expected that each instance follows a pattern, where the labels are in the next lower level below the unique identifier of an instance. This does not happen in case of the POWER instance set, and that is why there is the **POWER 2010** input option. It is also expected that the instances have no code that make them difficult to identify in the instance set, and that all of them have labels. This does not happen in case of the OAEI 2012 instance sets, and that is why there is the **OAEI 2012** input option. The third limitation concerns the location of the label of an instance, in the .NT and .TTL instance sets extensions. It is expected that the labels be right next of the / character. For example: <http://pt.wikipedia.org/wiki/**António_Guterres**><><><>.

3.6.6 Web Tool Screenshots

In this subsection, screenshots of the web tool are presented.

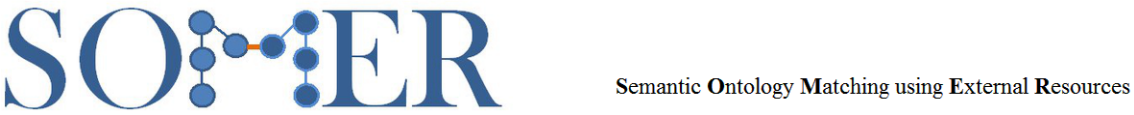


Figure 3.9: Web tool header screenshot

In this Figure, it is exhibited the header of the web tool, which is present in both input and output pages. The header is composed by the SOMER logo, which is the name of the project that focuses on several areas where is included instance matching, and to which I belong as well. By clicking on the logo, the SOMER official website[4] will be opened in a new tab. In the bottom right of the header, the meaning of the acronym is written out.

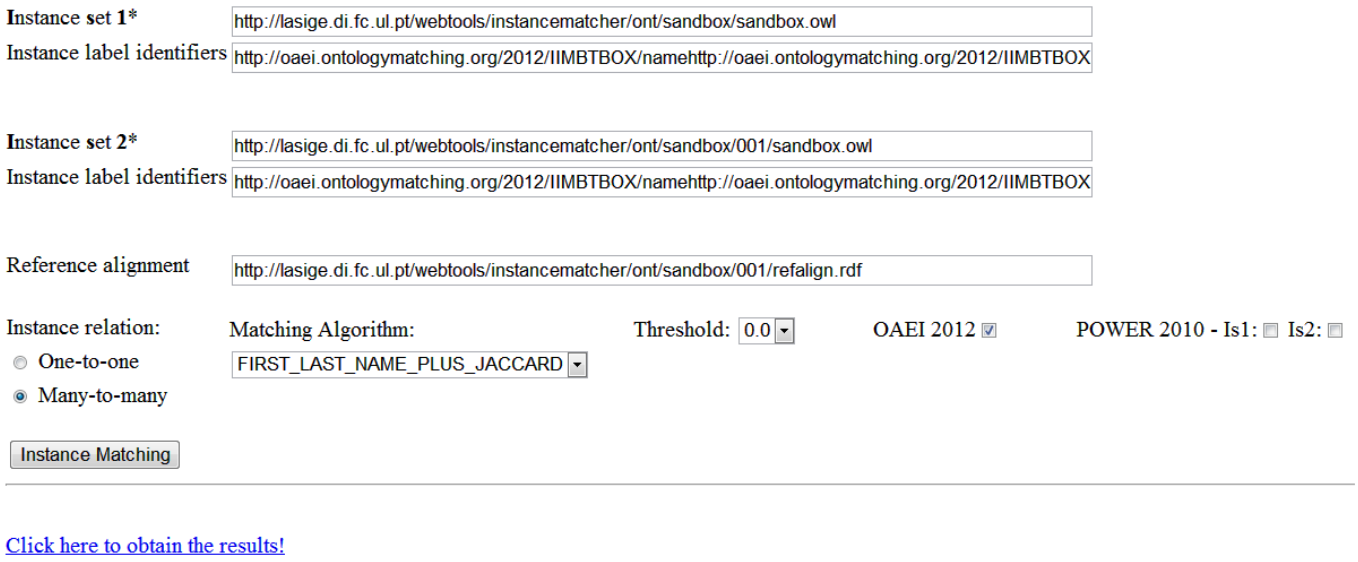


Figure 3.10: Web tool input screenshot. The image shows the input and their default values. At the bottom, the link to the output page, which appears post-submission.

[4] http://somer.fc.ul.pt/

Successful Scenario

Here, it is presented the successful scenario of the submission above.

Figure 3.11: Web tool output screenshot - part 1. The link corresponds to the final alignment produced.

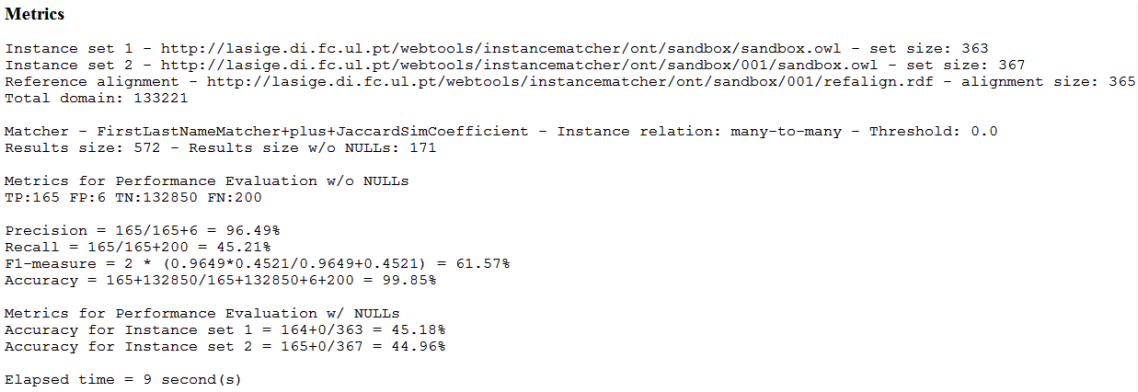


Figure 3.12: Web tool output screenshot - part 2. The image shows the panel containing the information about the input and the elapsed time of the all process. And because a reference alignment was given, is also displayed information about the final alignment assessment. Note: the *Results size: X* information is referring to the number of matches present in the final alignment; and the *Results size w/o Nulls: X* information is just referring to the number of pair of instances matched, present in the final alignment, without counting the instances matched to NULL.

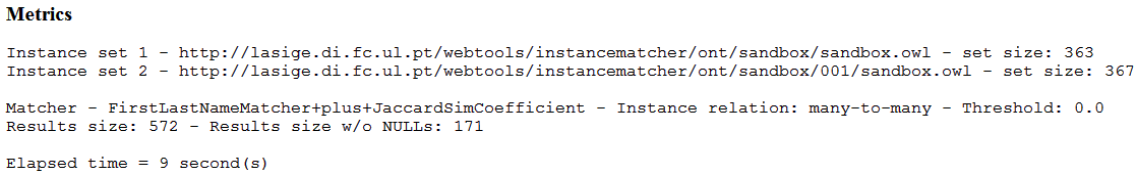


Figure 3.13: Web tool output screenshot - alternative. The image shows the panel containing the information about the input and the elapsed time of the all process. For the cases when no reference alignment is given, is just displayed summarized information about the final alignment. Note: the *Results size: X* information is referring to the number of matches present in the final alignment; and the *Results size w/o Nulls: X* information is just referring to the number of pair of instances matched, present in the final alignment, without counting the instances matched to NULL.

The image below, displays the sample of the final alignment produced. It is exhibited a pair of instances matched, and an instance matched to NULL, and their respective confidence score. The final alignment also gives the URI of the instance sets involved in the instance matching process. This information can be located at the beginning or at the end of the final alignment.

```
<Alignment>
 <map>
    <Cell>
      <entity2 rdf:resource="http://oaei.ontologymatching.org/2012/
      IIMBDATA/en/item3528632717660394758"/>
      <entity1 rdf:resource="http://oaei.ontologymatching.org/2012/
      IIMBDATA/en/fiji"/>
      <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
      1.0</measure>
      <relation>=</relation>
    </Cell>
  </map>
  <map>
    <Cell>
      <entity2 rdf:resource="http://oaei.ontologymatching.org/2012/
      IIMBDATA/en/item42699735707099327774"/>
      <entity1 rdf:resource="http://NULL/15367030870144571"/>
      <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
      0.6666666</measure>
      <relation>=</relation>
    </Cell>
  </map>
  </Alignment>
<Input>
  <InstanceSet2>http://lasige.di.fc.ul.pt/webtools/instancematcher/
  ont/sandbox/001/sandbox.owl</InstanceSet2>
  <InstanceSet1>http://lasige.di.fc.ul.pt/webtools/instancematcher/
  ont/sandbox/sandbox.owl</InstanceSet1>
</Input>
```

Exceptions

Here, it is presented some of the error messages displayed in the web tool, when some exceptions occur.

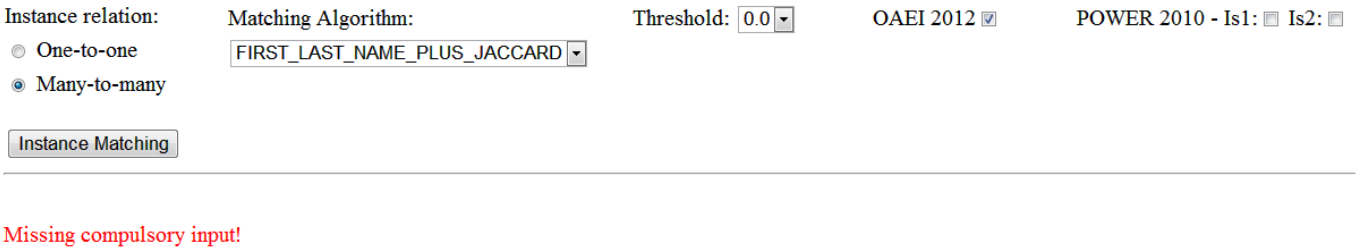


Figure 3.14: Missing compulsory input. This error message is displayed in the input page, if any of the mandatory input are left empty.

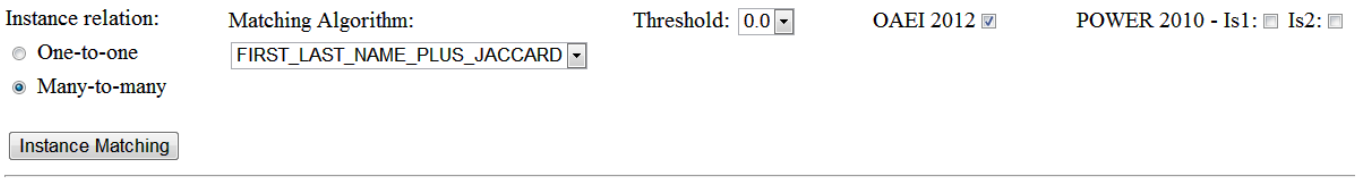


Figure 3.15: Input length violation. This error message is displayed in the input page, if any of the input length constraints are not respected.

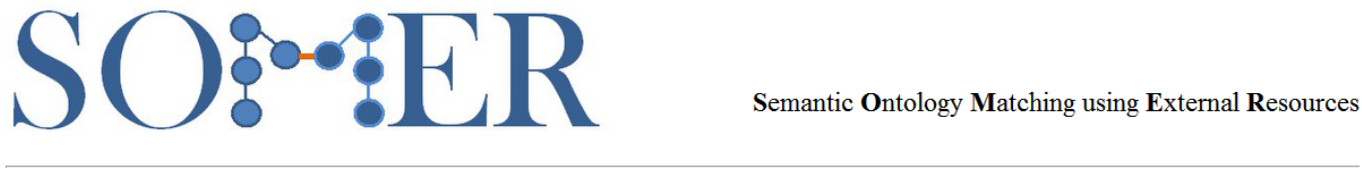
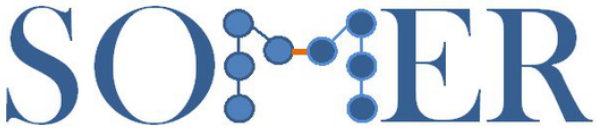


Figure 3.16: Instance matching execution error. This error message is displayed in the output page, if some error occurs during the instance matching process.

Figure 3.17: Alignment assessment error. This error message is displayed in the output page, if some error occurs during the alignment assessment process.



Semantic Ontology Matching using External Resources

Keep refreshing the page until the results appear, although the page auto-refreshes every 20 seconds

Error on Instance Matching! - Instance sets too big!

Figure 3.18: Instance sets too big error. This error message is displayed in the output page, if the instance sets introduced have to much instances for the web tool to handle.

Software

The Instance Matcher Web tool is hosted in the Server Chronos[5], that belongs to LaSIGE, with Linux as operating system, and Apache as web server.

The tool's web pages were written in HTML/CSS. The PHP language was used to collect the user input and to generate dynamic web page contents. PHP and HTML were also used to implement security measures in the Web tool.

The modules described in the above sections, were written in Java programming language using the Eclipse platform, and they were then incorporated in the Web tool through a .JAR (archive file format). Here, the PHP code was used to pass the user input as arguments to the .JAR, and to retrieve and to process its output.

[5]http://xldb.fc.ul.pt/wiki/XLDB_Servers_Internal_Pages

Chapter 4

Results

This chapter presents the results obtained by the system presented in the previous Chapter, in the following matching problems: OAEI 2012; POWER-DBpediaPT; POWER-Verbetes; and POWER-POWER alignments. All the alignments and metrics produced are available through this link: `http://lasige.di.fc.ul.pt/webtools/instancematcher/dissertation_work.zip`

4.1 OAEI 2012

This section describes the results of the OAEI 2012. The results are related with two tasks of the OAEI 2012 instance matching track[1]: the Sandbox task, which is composed by eleven instance sets; and the IIMB task, which is composed by eighty instance sets. To produce the results (many-to-many alignments), it was used the stratified 10-fold cross-validation machine learning technique. Thus, it was necessary to choose a classifier, in order to create a model capable of predicting if a given pair of instances match or not. The Weka software (Hall et al., 2009) provides several classifiers and, it was chosen some of them in order to test which would produce the best results, and also to choose which one would be used in the Web tool. The classifiers tested, encompassed some groups of classifiers, for the test to be as broad as possible: **Trees** - J48, RandomForest and RandomTree; **Meta** - RotationForest; **Bayes** - NaiveBayes; **Functions** - SMO; **Neural Networks** - MLP (**M**ulti**l**ayer **P**erceptron). It was also used the FirstLastNamePlusJaccard matcher as baseline.

The quality of the results, presented below, are expressed in **P**recision, **F**-measure and **R**ecall metrics. Because, the Sandbox and IIMB tasks have many instance sets the results were averaged, and in the case of the IIMB task, the eighty instance sets were split into four parts (001-020; 021-040; 041-060; 061-080). In the graphics, each dot of the baseline and of the classifiers tested corresponds to each part, but not necessarily in the order presented in the tables. All this strategy of presenting the results, graphics and tables, was based on the paper (Aguirre et al., 2012).

[1] http://oaei.ontologymatching.org/2012/

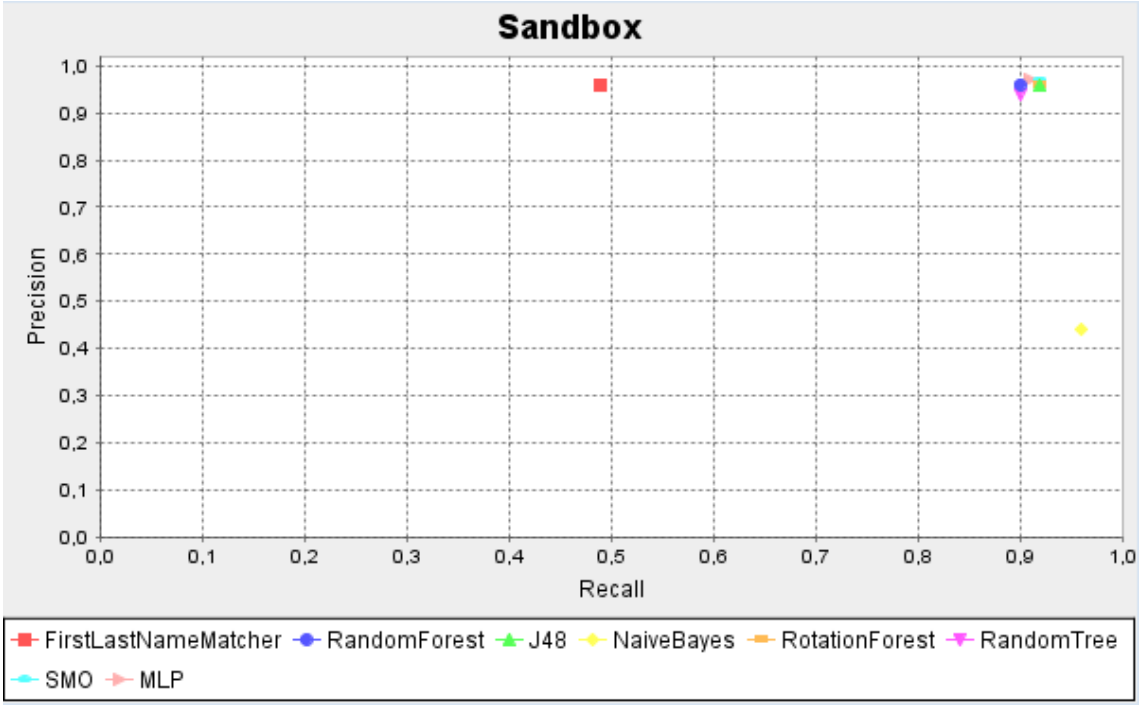


Figure 4.1: Precision/recall results of the Sandbox task

Test	001-011		
	P	F	R
FirstLastNamePlusJaccard	.96	.65	.49
RandomForest	.96	.93	.90
J48	.96	.94	.92
NaiveBayes	.44	.60	.96
RotationForest	.96	.94	.92
RandomTree	.94	.92	.90
SMO	.97	.94	.92
MLP	.97	.94	.91

Table 4.1: Results of the Sandbox task

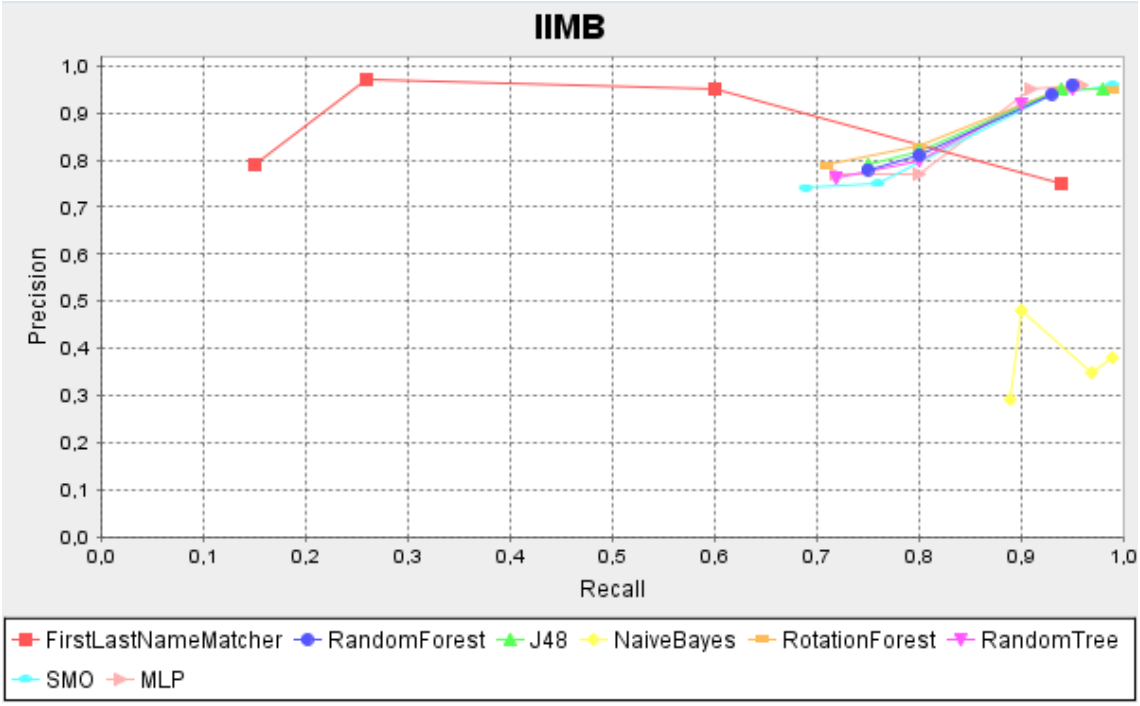


Figure 4.2: Precision/recall results of the IIMB task

Test	001-020			021-040			041-060			061-080		
	P	F	R	P	F	R	P	F	R	P	F	R
FirstLastNamePlusJaccard	.95	.74	.60	.97	.41	.26	.75	.83	.94	.79	.25	.15
RandomForest	.94	.94	.93	.96	.96	.95	.81	.81	.80	.78	.77	.75
J48	.95	.95	.94	.95	.97	.98	.82	.82	.80	.79	.77	.75
NaiveBayes	.35	.51	.97	.38	.55	.99	.48	.63	.90	.29	.44	.89
RotationForest	.95	.95	.94	.95	.97	.99	.83	.82	.80	.79	.75	.71
RandomTree	.92	.91	.90	.95	.95	.95	.80	.80	.80	.76	.74	.72
SMO	.94	.94	.93	.96	.98	.99	.75	.76	.76	.74	.71	.69
MLP	.95	.93	.91	.96	.96	.96	.77	.79	.80	.77	.74	.72

Table 4.2: Results of the IIMB task

4.1.1 Attributes selection

The results presented below, concern the attributes selection of the set of attributes chosen for the stratified 10-fold cross-validation technique, used in the OAEI 2012 matching problem. The attributes selection aimed at choosing the most relevant attributes for the production of the results. In each one of the 10 iterations performed in the stratified 10-fold cross-validation technique, it was applied together in the training set the `CfsSubsetEval`[2] and the `GreedyStepwise`[3] algorithms, provided by Weka, to select the attributes that would be used to train the classifier, and therefore to create the

[2]http://weka.sourceforge.net/doc.dev/weka/attributeSelection/CfsSubsetEval.html
[3]http://weka.sourceforge.net/doc.dev/weka/attributeSelection/GreedyStepwise.html

model capable of making the instance matching predictions. Note: to avoid the tampering of the dataset, the Attributes Selection technique is applied in a copy of the training set. This precaution is necessary because, in the stratified 10-fold cross-validation technique, the dataset is used 10 times corresponding to the 10 iterations.

In the graphics and tables below, is presented the number of times that each attribute was selected and the results produced by them.

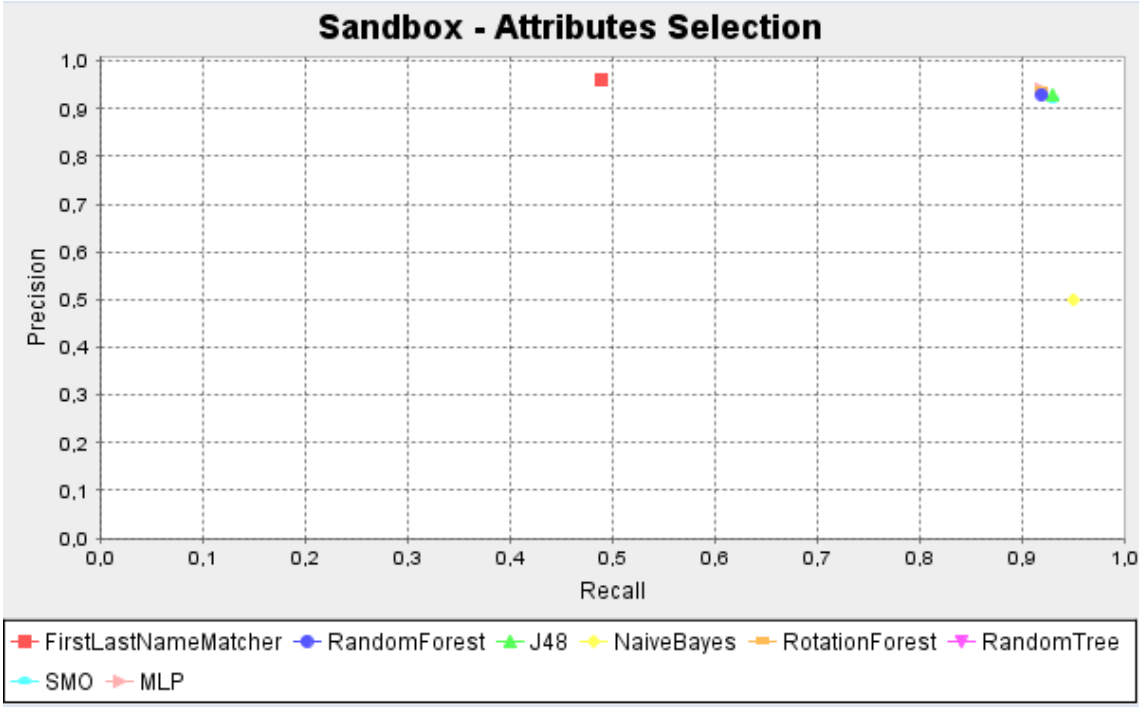


Figure 4.3: Precision/recall results of the Sandbox task - Attributes selection

Test	001-011		
	P	F	R
FirstLastNamePlusJaccard	.96	.65	.49
RandomForest	.93	.93	.92
J48	.93	.93	.93
NaiveBayes	.50	.66	.95
RotationForest	.94	.93	.92
RandomTree	.93	.93	.92
SMO	.92	.93	.93
MLP	.94	.93	.92

Table 4.3: Results of the Sandbox task - Attributes selection

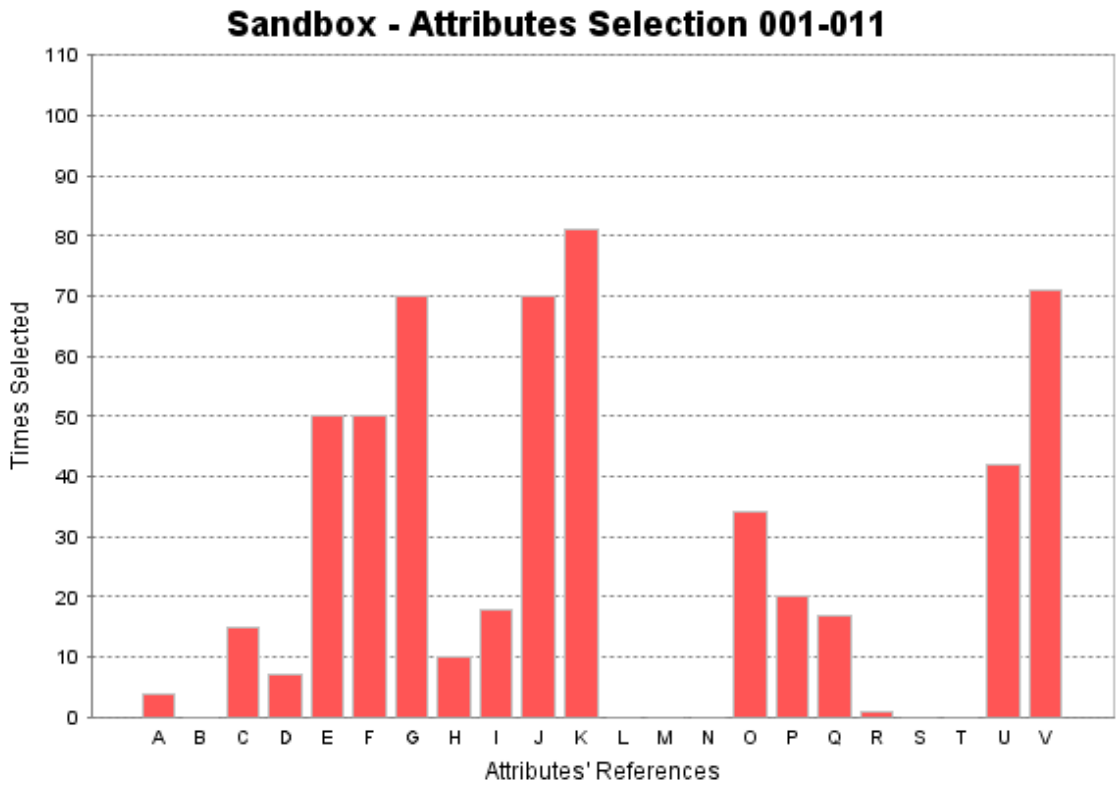


Figure 4.4: Times Selected/Attributes' References of the Sandbox task - Attributes selection

Attributes' References	Attributes' Names	Times Selected
A	idfSim1	4
B	topIdfSim2	0
C	idfSim3	15
D	topIdfSim4	7
E	cosSim5	50
F	idfSim6	50
G	topIdfSim7	70
H	editSim8	10
I	countSim9	18
J	countSim10	70
K	countSim11	81
L	name1Len	0
M	name2Len	0
N	firstSame	0
O	firstSameEC	34
P	lastSame	20
Q	lastSameEC	17
R	twoLastSame	1
S	twoLastSameEC	0
T	firstLastSame	0
U	firstLastSameEC	42
V	JCValue	71

Table 4.4: Table showing the times each attribute was selected, and their respective references. Sandbox task - Attributes selection

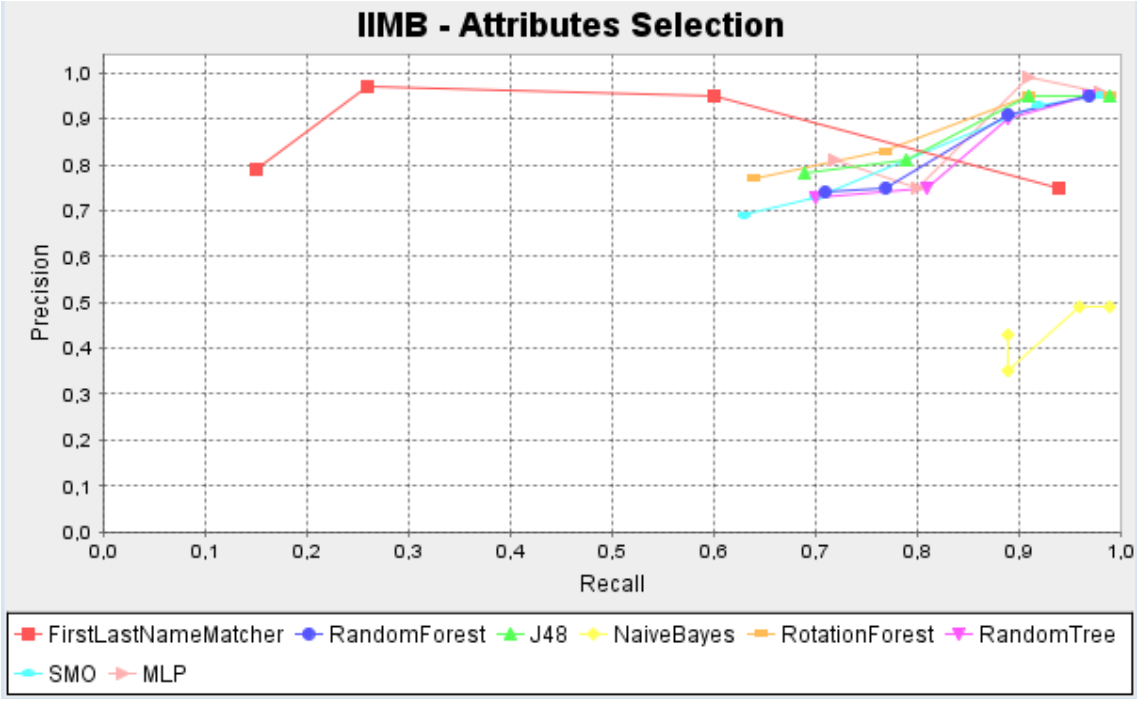


Figure 4.5: Precision/recall results of the IIMB task - Attributes selection

Test	001-020			021-040			041-060			061-080		
	P	F	R	P	F	R	P	F	R	P	F	R
FirstLastNamePlusJaccard	.95	.74	.60	.97	.41	.26	.75	.83	.94	.79	.25	.15
RandomForest	.91	.90	.89	.95	.96	.97	.75	.76	.77	.74	.73	.71
J48	.95	.93	.91	.95	.97	.99	.81	.80	.79	.78	.73	.69
NaiveBayes	.49	.65	.96	.49	.66	.99	.43	.58	.89	.35	.50	.89
RotationForest	.95	.93	.91	.95	.97	.99	.83	.80	.77	.77	.70	.64
RandomTree	.90	.90	.89	.95	.96	.97	.75	.78	.81	.73	.72	.70
SMO	.93	.93	.92	.95	.97	.98	.73	.72	.70	.69	.66	.63
MLP	.99	.95	.91	.96	.97	.98	.75	.77	.80	.81	.76	.72

Table 4.5: Results of the IIMB task - Attributes selection

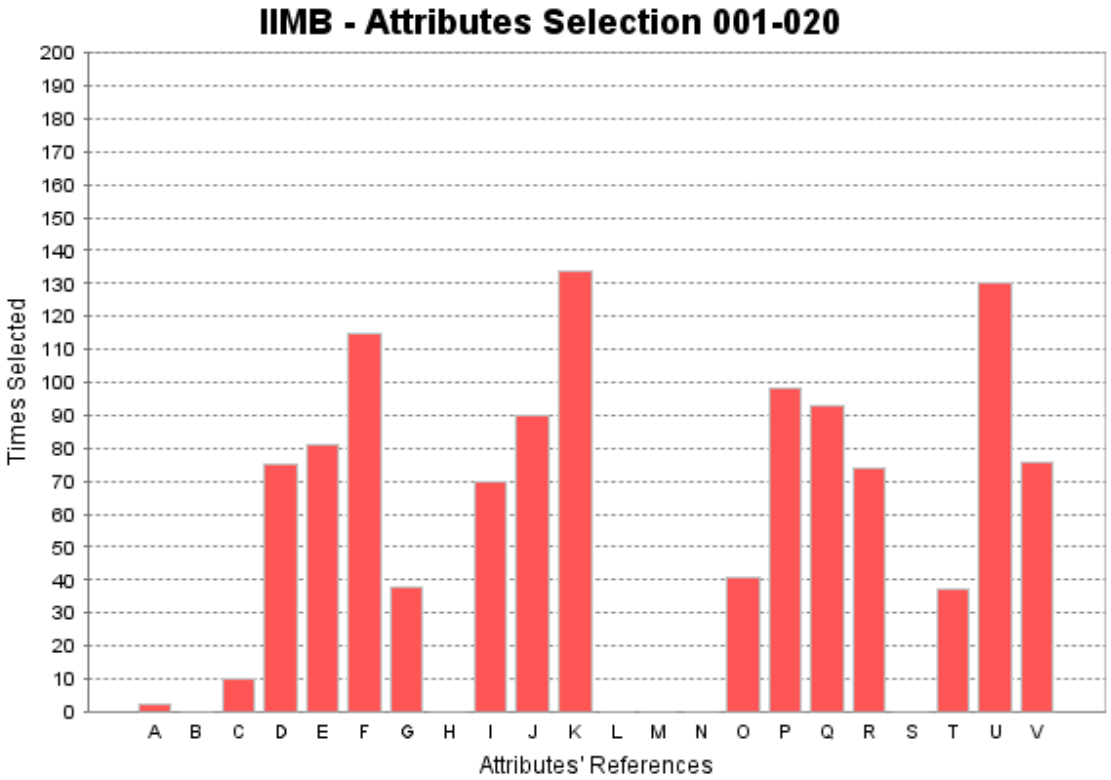


Figure 4.6: Times Selected/Attributes' References of the IIMB 001-020 - Attributes selection

Attributes' References	Attributes' Names	Times Selected
A	idfSim1	2
B	topIdfSim2	0
C	idfSim3	10
D	topIdfSim4	75
E	cosSim5	81
F	idfSim6	115
G	topIdfSim7	38
H	editSim8	0
I	countSim9	70
J	countSim10	90
K	countSim11	134
L	name1Len	0
M	name2Len	0
N	firstSame	0
O	firstSameEC	41
P	lastSame	98
Q	lastSameEC	93
R	twoLastSame	74
S	twoLastSameEC	0
T	firstLastSame	37
U	firstLastSameEC	130
V	JCValue	76

Table 4.6: Table showing the times each attribute was selected, and their respective references. IIMB 001-020 - Attributes selection

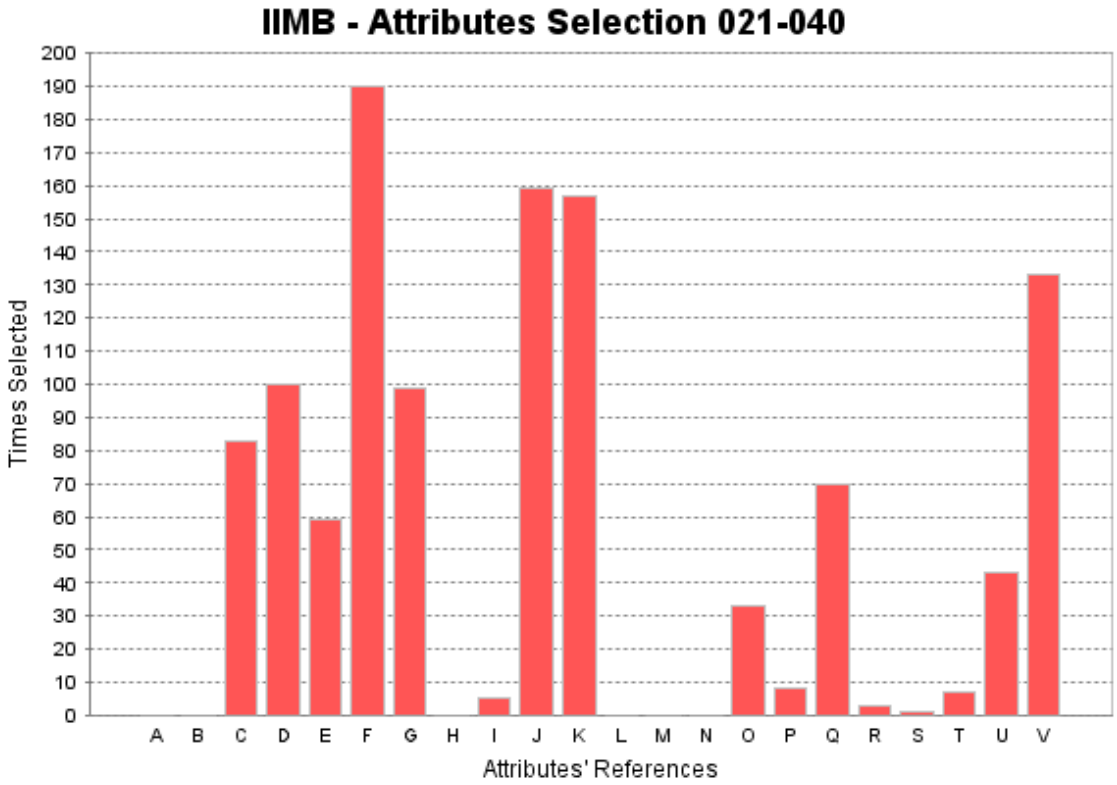


Figure 4.7: Times Selected/Attributes' References of the IIMB 021-040 - Attributes selection

Attributes' References	Attributes' Names	Times Selected
A	idfSim1	0
B	topIdfSim2	0
C	idfSim3	83
D	topIdfSim4	100
E	cosSim5	59
F	idfSim6	190
G	topIdfSim7	99
H	editSim8	0
I	countSim9	5
J	countSim10	159
K	countSim11	157
L	name1Len	0
M	name2Len	0
N	firstSame	0
O	firstSameEC	33
P	lastSame	8
Q	lastSameEC	70
R	twoLastSame	3
S	twoLastSameEC	1
T	firstLastSame	7
U	firstLastSameEC	43
V	JCValue	133

Table 4.7: Table showing the times each attribute was selected, and their respective references. IIMB 021-040 - Attributes selection

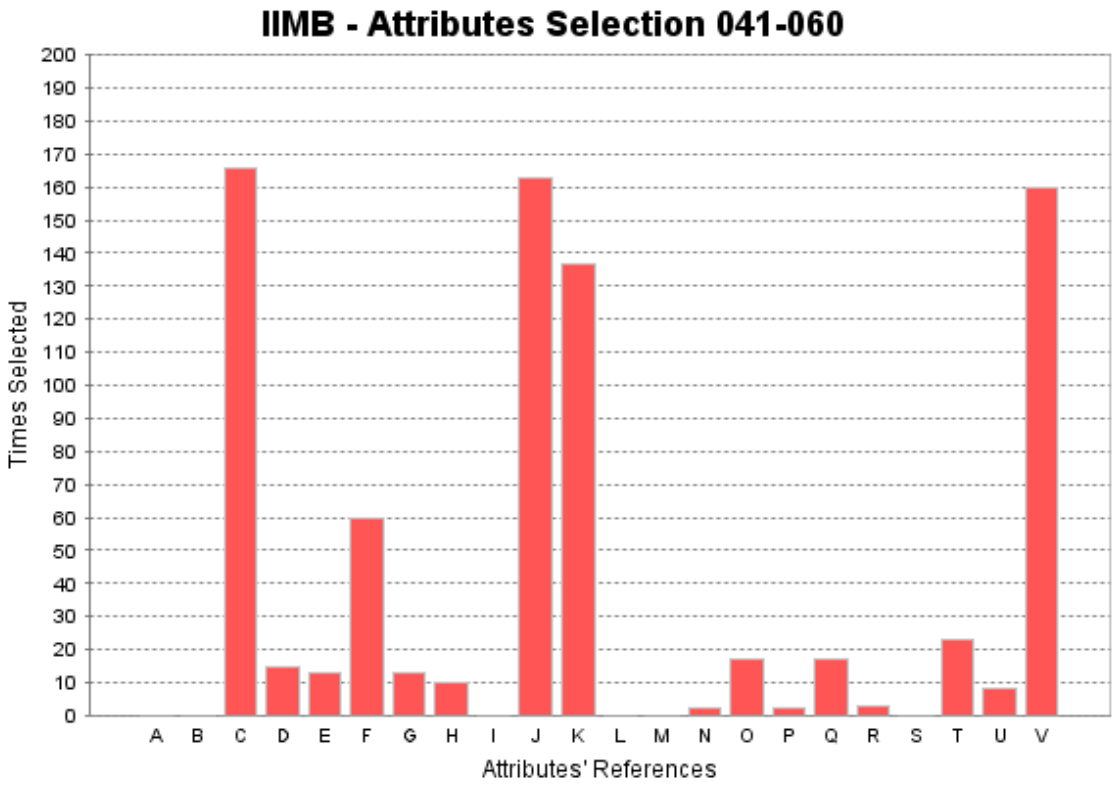


Figure 4.8: Times Selected/Attributes' References of the IIMB 041-060 - Attributes selection

Attributes' References	Attributes' Names	Times Selected
A	idfSim1	0
B	topIdfSim2	0
C	idfSim3	166
D	topIdfSim4	15
E	cosSim5	13
F	idfSim6	60
G	topIdfSim7	13
H	editSim8	10
I	countSim9	0
J	countSim10	163
K	countSim11	137
L	name1Len	0
M	name2Len	0
N	firstSame	2
O	firstSameEC	17
P	lastSame	2
Q	lastSameEC	17
R	twoLastSame	3
S	twoLastSameEC	0
T	firstLastSame	23
U	firstLastSameEC	8
V	JCValue	160

Table 4.8: Table showing the times each attribute was selected, and their respective references. IIMB 041-060 - Attributes selection

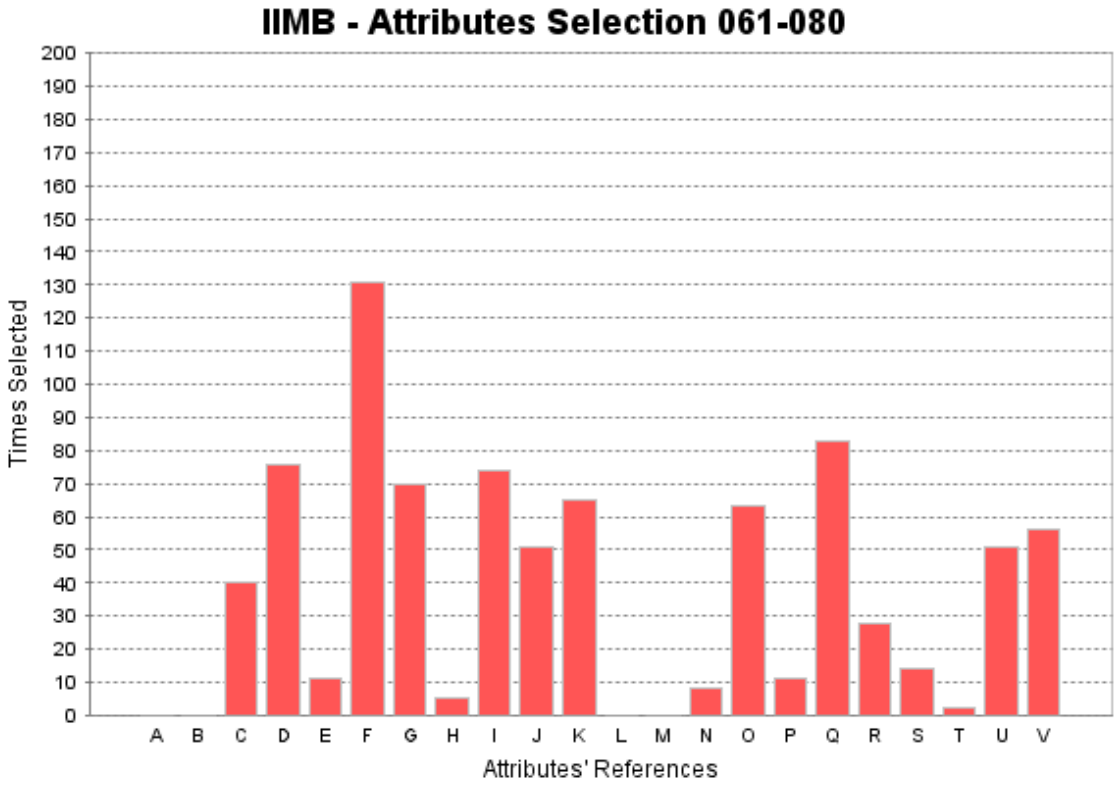


Figure 4.9: Times Selected/Attributes' References of the IIMB 061-080 - Attributes selection

Attributes' References	Attributes' Names	Times Selected
A	idfSim1	0
B	topIdfSim2	0
C	idfSim3	40
D	topIdfSim4	76
E	cosSim5	11
F	idfSim6	131
G	topIdfSim7	70
H	editSim8	5
I	countSim9	74
J	countSim10	51
K	countSim11	65
L	name1Len	0
M	name2Len	0
N	firstSame	8
O	firstSameEC	63
P	lastSame	11
Q	lastSameEC	83
R	twoLastSame	28
S	twoLastSameEC	14
T	firstLastSame	2
U	firstLastSameEC	51
V	JCValue	56

Table 4.9: Table showing the times each attribute was selected, and their respective references. IIMB 061-080 - Attributes selection

Discussion

In this section it is discussed why, in the Attributes Selection task, some attributes were selected more times than others, based on the analysis made on some datasets that were produced in this task.

In the attributes based on the paper (Rong et al., 2012), the most selected attributes were: *double idfSim3; double topIdfSim4; double cosSim5; double idfSim6; double topIdfSim7; double countSim10; double countSim11*. In my opinion, because their double values were, most of the times, greater than 0.0. The attributes that were selected few times: *double idfSim1; double editSim8; double countSim9*. In my opinion, because their double values were few times greater than 0.0. The attributes that were never selected: *double topIdfSim2*. In my opinion, because its double value was almost always 0.0.

Concerning the attributes proposed by the REACTION group. The most selected attributes were: *double firstSameEC; double lastSameEC; double firstLastSameEC; double JCvalue*. And the attributes that were selected few times: *boolean firstSame; boolean lastSame; boolean firstLastSame; boolean twoLastSame; double twoLastSameEC*. These two groups reveal a pattern where the attributes selection algorithms seemed to have preferred the double type attributes rather than the boolean type ones because, in my opinion, the double type provides more detailed information (it provides real number values). The attributes that were never selected: *int name1Len; int name2Len*. In my opinion, because the integer values do not provide so much detailed information comparing with the double values.

The *int name1Len* and the *int name2Len* attributes were never used, and this situation set the decision to not use them in the set of attributes, that was chosen to create the instance matching predictive model, that is used by the Machine Learning matcher.

4.1.2 Uniform Distribution

The usage of the stratified 10-fold cross-validation technique implies the creation of a dataset where training and test sets can be extracted from. In this work, the datasets are built from the instance sets that are to be (re) aligned. Furthermore, to assign to each entry of the dataset the categories of *yes* (match) or *no* (do not match), it is used the reference alignment between those instance sets. More precisely, let us consider the following example: on creating the entry related with the instance A_1 of the instance set 1 with the instance B_1 of the instance set 2, if this pair of instances is present in the reference alignment, it is assigned the *yes* category to the entry. This creates a situation, in the dataset, where there are many entries with the *no* category and few entries with the *yes* category. For example: on aligning the IIMB reference instance set, which contains 363 instances, with the IIMB 035 instance set, which contains 367 instances, is built a dataset of 133221 (363*367) entries, but only 365 of them (the number of pairs of instances present in the reference alignment) belong to the *yes* category.

To find out if this situation harms the quality of the predictive model, it was performed tests in the Sandbox task, where in the training set the *yes* and *no* categories are uniformly distributed, i.e., both of the categories have the same (or nearly the same) number of entries. The Weka software provides a set of mechanisms to do it, but only two of them

were used: `Resample`[4] - in which the category with the higher frequency is decreased, and the category with the lower frequency is increased (new entries are created from the existing ones), until they "meet in the middle" or nearly; and `SpreadSubsample`[5] - in which the category with the higher frequency is decreased until it reaches the frequency of the lower frequency category. Note: to avoid the tampering of the dataset and bias situations, i.e., having the same entry in the training and test sets at the same time, the Uniform Distribution technique is applied in a copy of the training set. This precaution is necessary because, in the stratified 10-fold cross-validation technique, the dataset is used 10 times corresponding to the 10 iterations.

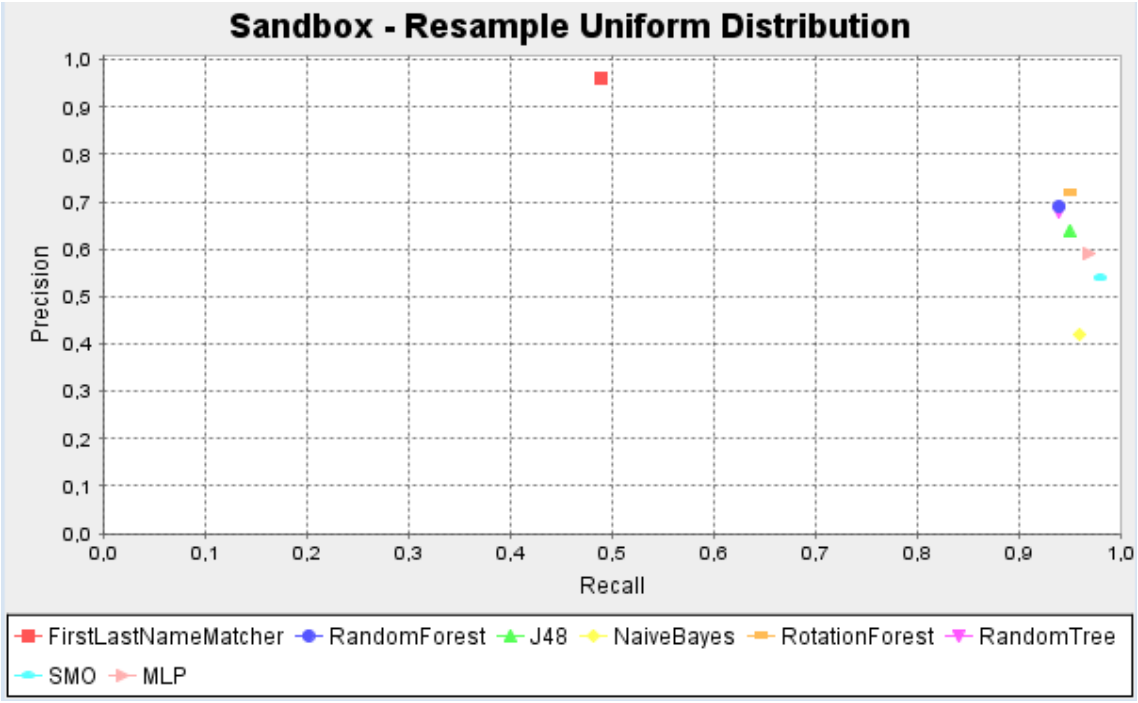


Figure 4.10: Precision/recall results of the Sandbox task - Resample Uniform Distribution

Test	001-011		
	P	F	R
FirstLastNamePlusJaccard	.96	.65	.49
RandomForest	.69	.80	.94
J48	.64	.77	.95
NaiveBayes	.42	.58	.96
RotationForest	.72	.82	.95
RandomTree	.68	.79	.94
SMO	.54	.70	.98
MLP	.59	.73	.97

Table 4.10: Results of the Sandbox task - Resample Uniform Distribution

[4]http://weka.sourceforge.net/doc.dev/weka/filters/supervised/instance/Resample.html
[5]http://weka.sourceforge.net/doc.dev/weka/filters/supervised/instance/SpreadSubsample.html

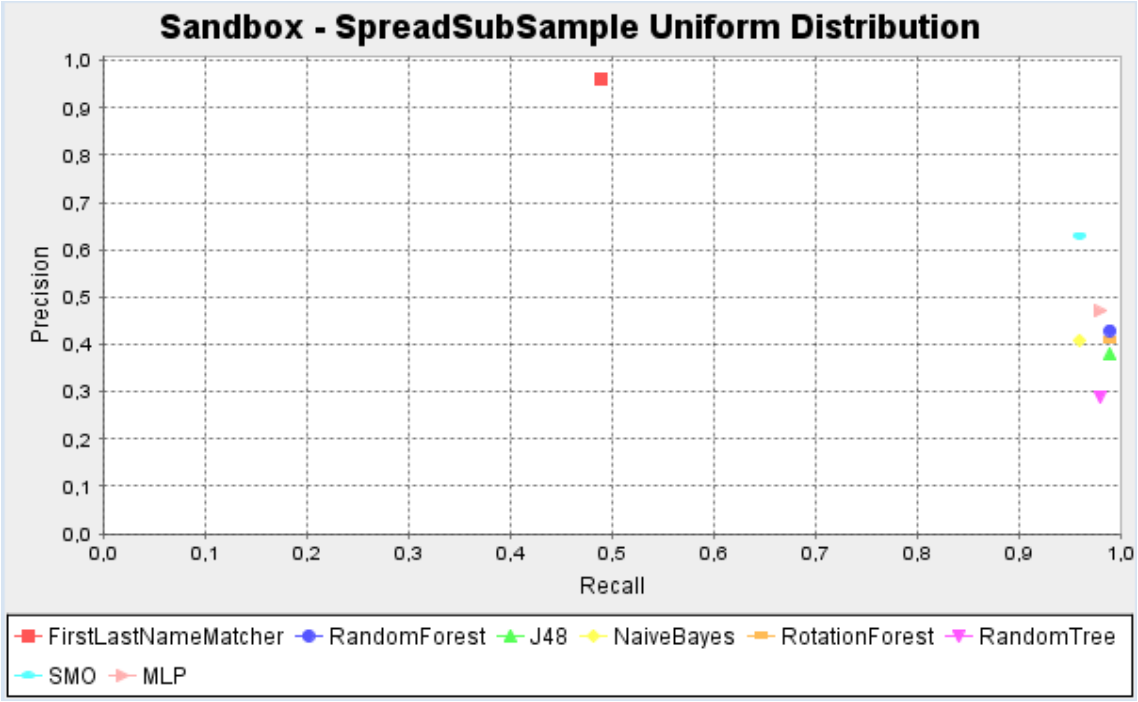


Figure 4.11: Precision/recall results of the Sandbox task - SpreadSubsample Uniform Distribution

Test	001-011		
	P	F	R
FirstLastNamePlusJaccard	.96	.65	.49
RandomForest	.43	.60	.99
J48	.38	.55	.99
NaiveBayes	.41	.58	.96
RotationForest	.41	.58	.99
RandomTree	.29	.45	.98
SMO	.63	.76	.96
MLP	.47	.64	.98

Table 4.11: Results of the Sandbox task - SpreadSubsample Uniform Distribution

The results presented above show that, although the Recall metric have increased, the Precision metric have highly decreased, comparing with the Sandbox results of the previous sections. This test, set the decision to not use the Uniform Distribution technique in the other matching problems presented next and also, to not use it in the predictive model incorporated in the Web tool.

Discussion

The results presented in this OAEI 2012 section, showed that the Rotation Forest classifier was one of the top classifiers, based on the F-measure metric. Therefore, it was chosen to create the predictive model that is incorporated in the Web tool, through the Machine Learning matcher.

4.2 POWER-DBpediaPT Alignment

This section presents the POWER-DBpediaPT alignment results. This matching problem aimed at producing an one-to-one alignment with 90% of Unilateral Accuracy at least, in the POWER side, although DBpediaPT Unilateral Accuracy was also calculated. Furthermore, it was also used to set which one-to-one matcher was to be used in the Web tool. There were considered three algorithms, that were already explained in the previous Chapter: (A) labels-sum+alphabetic-order; (B) confidence-score+attribute-sum; Hungarian Algorithm (Kuhn, 1955). The matchers used were the FirstLastNamePlusJaccard, and the Stratified 10-Fold Cross-validation machine learning technique, using as classifier the Random Forest.

As said before, the POWER instance set contains duplicate instances, i.e., two or more instances representing the same entity. To put all the one-to-one matchers in the same conditions, these duplicates were eliminated from the POWER-DBpediaPT reference alignment, and from the POWER instance set. Although, in this last case the elimination was performed by filtering out the duplicate instances from the instance matching process. The criteria to choose which one of the duplicates should be not considered was randomly set by me. For example, the entity X is represented twice in the POWER instance set, through the instances A_1 and A_2, that match to the instance B_1 in DBpediaPT. So, I randomly choose the instance A_2, to be filtered out from the POWER instance set, and to not be present in the reference alignment. Instead, in the reference alignment is present the pair (A_1, B_1).

The instances in POWER "eliminated" were only the ones that had a matching in the DBpediaPT. The duplicate instances that had no matching in the DBpediaPT (instances matching to NULL), were kept.

One-to-one Matcher	FirstLastNamePlusJaccard	Stratified 10-Fold Cross-validation
(A)	97.29% - 87.25%	99.11% - 95.97%
(B)	97.25% - 86.91%	99.18% - 96.14%
Hungarian	94.54% - 80.54%	99.04% - 95.81%

Table 4.12: POWER-DBpediaPT Alignment Results. This table shows the one-to-one and the instance matchers used in this matching problem. The results are exposed as: POWER Unilateral Accuracy - DBpediaPT Unilateral Accuracy.

According to the results presented in the Table 4.12, the (A) one-to-one matcher produced the best results, and therefore, it was chosen to be used in the Web tool. In the FirstLastNamePlusJaccard: (A) >(B) = (0,04+0,34) 0,38; and in the Stratified 10-Fold Cross-validation: (A) <(B) = (0,07+0,17) 0,24. Conclusion: (A)0,38 >(B)0,24.

4.3 POWER-Verbetes Alignment

This section presents the POWER-Verbetes alignment results. For this matching problem there was no reference alignment, and this situation required that the one-to-one alignment produced was partly assessed by the REACTION group[6], which is the final user of the alignment, and the other part by me, the author of the alignment. The matcher used was the Machine Learning, using as classifier the Rotation Forest.

```
<map>
    <Cell>
      <entity2 rdf:resource="http://NULL/15162729615603"/>
      <entity1 rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#jose_eduardo_martins"/>
      <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.8818546</measure>
      <relation>=</relation>
    </Cell>
</map>
<map>
    <Cell>
      <entity2 rdf:resource="http://services.sapo.pt/InformationRetrieval/Verbetes/WhoIs?name=José Sócrates"/>
      <entity1 rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#jose_socrates"/>
      <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.9816217</measure>
      <relation>=</relation>
    </Cell>
</map>
<map>
    <Cell>
      <entity2 rdf:resource="http://services.sapo.pt/InformationRetrieval/Verbetes/WhoIs?name=António Lobo Antunes"/>
      <entity1 rdf:resource="http://NULL/15213734326606"/>
      <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.98645014</measure>
      <relation>=</relation>
    </Cell>
</map>
```

Figure 4.12: POWER-Verbetes Alignment sample

In the alignment, the REACTION group looked for the most well-known Portuguese politicians and attested that they were all correctly matched. But this feedback was not enough, because it comprehended few Portuguese politicians (7 in total). That is why I looked for more. I took 32 pairs of instances for which was identified a match, and just 3 of them were wrong. And I also took 20 instances matching to NULL, and they were all correct. I assume that, the alignment, by the number of matches assessed, has high likelihood of being mostly correct.

[6]http://dmir.inesc-id.pt/project/Reaction

4.4 POWER-POWER Alignment

As said before, the POWER instance set has duplicate instances. To help the REACTION group to find these instances, a many-to-many alignment between the POWER itself was produced, and provided to the group. The purpose was to allow to an instance to be matched by multiple instances (including itself), in order to find the duplicates. The matcher used was the Machine Learning, using as classifier the Rotation Forest.

```
<map>
  <Cell>
   <entity2 rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#jose_manuel_dias_custodio"/>
   <entity1 rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#jose_manuel_dias_custodio"/>
   <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.979924</measure>
   <relation>=</relation>
  </Cell>
</map>
<map>
  <Cell>
   <entity2 rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#jose_manuel_dias_custodio"/>
   <entity1 rdf:resource="http://dmir.inesc-id.pt/pub/publications/2010/power#jose_m._dias_custodio"/>
   <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.9792201</measure>
   <relation>=</relation>
  </Cell>
</map>
```

Figure 4.13: POWER-POWER Alignment sample. The Figure shows an instance aligned with itself, and also with another instance.

Chapter 5

Conclusion

In this chapter is discussed the results achieved in this work, within the scope of the objectives presented in Section 1.2.

Development of instance matching algorithms In this objective, it was implemented three instance matching algorithms: FirstLastNamePlusJaccard matcher; Machine Learning matcher; and the Stratified 10-fold Cross-validation matcher. Moreover, it was also implemented algorithms that make part of the instance matching process, such as: the pre-processing module, where it is extracted the literal information belonging to each instance; the pre-processing sub-modules, where it is applied a cleaning process in the literal information; and also the implementation of the Hungarian Algorithm, and the creation and implementation of the algorithms (A) labels-sum+alphabetic-order and (B) confidence-score+attribute-sum;

Real world instance alignments

POWER-DBpediaPT instance alignments This instance matching problem was difficult to fulfil due to the duplicate instances of POWER. But the experiments made in it, set the decision to choose the (A) labels-sum+alphabetic-order one-to-one matcher to be used in the Web tool. Using this one-to-one matcher, in the First-LastNamePlusJaccard matcher, the POWER Unilateral Accuracy achieved 97.29%, and the DBpediaPT Unilateral Accuracy achieved 87.25%; and in the Stratified 10-Fold Cross-validation, the POWER Unilateral Accuracy achieved 99.11%, and the DBpediaPT Unilateral Accuracy achieved 95.97%. The reference alignment (one-to-one) used in this matching problem was built by me, which I provided to the REACTION group;

POWER-Verbetes instance alignments The one-to-one alignment produced by the Machine Learning matcher was provided to the REACTION group, and because there was no reference alignment, the alignment produced was partly assessed by the REACTION group, which is the final user of the alignment, and the other part by me, the author of the alignment. The assessment was positive from both parts;

POWER-POWER instance alignments The many-to-many alignment produced by the Machine Learning matcher was provided to the REACTION group, to make easier for them to identify the duplicate instances.

OAEI 2012 Although this matching problem was not an objective, I am very pleased about this task, because its tests were reference to make decisions about other subjects in this work. Namely: the choice of the Rotation Forest classifier, based on the results of the F-measure metric, to create the predictive model that is incorporated in the Web tool, through the Machine Learning matcher; the choice to not use the *name1Len* and *name2Len* attributes in the creation of the already mentioned predictive model; and the choice to not use the Uniform Distribution technique in the same model, and in the other instance matching problems. About the results presented in this instance matching problem, it is possible to say that the F-measure was higher, in most cases, in the Stratified 10-fold Cross-Validation matcher then in the FirstLastNamePlusJaccard matcher, that was used as baseline.

Evaluation Metrics To evaluate the alignments produced, it was implemented the Precision, Recall, F-measure and Accuracy metrics. Moreover, it was also created and implemented the Unilateral Accuracy metric that concerns not only the instances for which it was identified a match, but also the instances for each no match was identified, i.e., the instances matching to NULL. In this last case, it was implemented an algorithm to assign a confidence score for this type of match, which includes the instances that were completely excluded in the one-to-one matcher process, and therefore, match to NULL. The Unilateral Accuracy is calculated for each instance set.

Instance matcher Web tool development The Web tool incorporates the modules developed and the decisions made within the scope of the others objectives and tasks. From the "Development of instance matching algorithms" objective incorporates the FirstLastNamePlusJaccard and the Machine Learning matchers. And also, the pre-processing module and sub-modules. From the OAEI 2012, the way the predictive model, used by the Machine Learning matcher, was created: using the Rotation Forest classifier; without using *name1Len* and *name2Len* attributes as criteria; and without applying the Uniform Distribution technique. And from the POWER-DBpediaPT alignment, the (A) labels-sum+alphabetic-order one-to-one matcher. Moreover, the POWER and DBpediaPT instance sets provided the training set, that was used to build the predictive model. The Web tool also implements all the evaluation metrics developed, and allows the user to select the minimum confidence score of the matches to be included in the alignment produced to the output, through the Threshold option. Furthermore, it has options related with the OAEI 2012 and POWER.

In terms of usage, the Web tool has already default values for the users to try it in a single click of a button, but if they change any of them, the new values are preserved for the next usage. To allow the user to submit several instance matching operations, without waiting for each operation to finish, in each submission a link to the output page is displayed. It also supports several instance sets extensions: .RDF; .OWL; .NT; and .TTL. To avoid network time-outs, if the instance matching process takes too long, the results appear in the output page by refreshing it, although the page auto-refreshes every 20 seconds. In terms of feedback, this Web tool displays various messages in both successful and exception scenarios, where it is included security related messages. Concerning the security area, the purpose was to limit

the user input, in terms of number of characters, and to avoid the execution of shell or other (malicious) commands.

5.1 Future Work

Future work would include:

- OAEI 2014 participation. Using the experience acquired in the OAEI 2012;
- In the Web tool, the inclusion of an option where the user could choose one of the three one-to-one matchers developed by me;
- The increment of the capacity of the Web tool to support larger instance sets;
- The development of instance matching algorithms using structural approaches, as well. These techniques can be used to find correspondences between instances, by assessing their properties similarities (internal structure), and their relations similarities (relational structure), by comparing their relations with other instances;
- Writing a paper about the Instance Matcher Web tool.

Bibliography

Aguirre, J. L., B. C. Grau, K. Eckert, J. Euzenat, A. Ferrara, R. W. van Hague, L. Hollink, E. Jimenez-Ruiz, C. Meilicke, A. Nikolov, et al. (2012). Results of the ontology alignment evaluation initiative 2012. In *Proc. 7th ISWC workshop on ontology matching (OM)*, pp. 73–115.

Berners-Lee, T., J. Hendler, O. Lassila, et al. (2001). The semantic web. *Scientific american 284*(5), 28–37.

Bizer, C., T. Heath, and T. Berners-Lee (2009). Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS) 5*(3), 1–22.

Breiman, L. (2001). Random forests. *Machine Learning 45*(1), 5–32.

Couto, F. and H. Pinto (2013). The next generation of similarity measures that fully explore the semantics in biomedical ontologies. *Journal of Bioinformatics and Computational Biology 11*(5 (1371001)), 1–12.

Couto, F., M. Silva, and P. Coutinho (2005). Finding genomic ontology terms in text using evidence content. *BMC Bioinformatics 6*(S1(S21)), 1–6.

Ehrig, M. and Y. Sure (2005). Foam-framework for ontology alignment and mapping-results of the ontology alignment evaluation initiative. In *Workshop on integrating ontologies*, Volume 156, pp. 72–76.

Euzenat, J. and P. Shvaiko (2007). *Ontology matching*. Heidelberg (DE): Springer-Verlag.

Gruber, T. (2008). What is an ontology. *Encyclopedia of Database Systems 1*.

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten (2009). The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter 11*(1), 10–18.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. 1. *New phytologist 11*(2), 37–50.

John, G. H. and P. Langley (1995). Estimating continuous distributions in bayesian classifiers. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, San Mateo, pp. 338–345. Morgan Kaufmann.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly 2*(1-2), 83–97.

Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice Hall PTR.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, Volume 10, pp. 707.

Moreira, S., D. Batista, P. Carvalho, F. Couto, and M. Silva (2012). Tracking politics with POWER. *Program: electronic library and information systems in press*.

Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola (Eds.), *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers.

Rodriguez, J. J., L. I. Kuncheva, and C. J. Alonso (2006). Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*(10), 1619–1630.

Rong, S., X. Niu, E. W. Xiang, H. Wang, Q. Yang, and Y. Yu (2012). A machine learning approach for instance matching based on similarity metrics. In *The Semantic Web–ISWC 2012*, pp. 460–475. Springer.

Rosenblatt, F. (1961). Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, DTIC Document.